The design of a software program for the simulation of a two-dimensional model of the knee-joint

Report number 92.013 A

by: Niek van Nunen
Frank Peters

Eindhoven University of Technology, December 1991

Summary

This paper describes the design of a software program for the simulation of a two-dimensional model of the knee-joint. The purpose of the program is to train students in using and understanding the basics of parameter variation.
The joint is seen as a simple planar four bar mechanism with rigid links.
The analysis is divided into three parts:
First an exact problem definition is defined, followed by a graphical analysis. This graphical analysis is again the basis for the mathematical analysis, the third part.
With the results of the mathematical analysis the software program is written, which calculates the centrodes of the mechanism, and given the tibial contact profile (a straight line or a parabola), it also calculates the femoral contact profile.
Furthermore it is possible to calculate the effects of the variations of parameters, and a diagram which shows the slip ratio as a function of the flexion angle is drawn.
The most important reason to conclude that the model is correct, is the written subroutine MOVIE, which draws the mechanism and the four curves in several positions.
Finally a short description of the program is given, as well as the program text itself.
It is recommended to run the program on at least a 80286 PC/AT with coprocessor. Also a colour graphics card with colour monitor is recommended. The last appendix contains a user's manual.

1. Contents

1. Contents **1**
2. Symbols **3**
3. Preface **4**
4. Introduction **5**
5. Strategy **7**
6. Adjusted problem definition **7**
7. Global Method: A graphical analysis **8**
7.1 The flexion angle γ 8
7.2 The input 8
7.3 Construction of the centrodes 9
7.4 Construction of the femoral contact profile 10
8. Mathematical model **12**
8.1 The centrodes 12
8.1.1 Definitions 13
8.1.1.1 The angle $\varphi(\alpha)$ 13
8.1.1.2 The angle $\theta(\alpha)$ 14
8.1.2 Interpretation of results 15
8.1.3 The tibial centrode 15
8.1.4 The femoral centrode 17
8.2 The femoral contact profile 19
8.2.1 The distance l 19
8.2.1.1 Tibial contact profile is a straight line 21
8.2.1.2 Tibial contact profile is a parabola 22
8.2.2 The angle χ_i 23
8.2.3 The construction of the femoral contact profile 24
9. The programming **26**
9.1 Program structure 26
9.2 Main program 27
9.3 Subroutines 27
9.3.1 Subroutine INPUT 27
9.3.2 Subroutine CENTRO 28
9.3.3 Subroutine FEMORA 29
9.3.4 Subroutine OUTPUT 30
9.3.5 Subroutine PARA 30
9.3.6 Subroutine SLIP 31
9.3.7 Subroutine MOVIE 32

10. Model validation **33**
10.1 Conditions resulting from the model 33
10.2 Symmetrical mechanism 33
10.3 The Perspex model 33
10.4 Subroutine MOVIE 34
11. Conclusions **35**
12. Literature **37**

Note: This report is accompanied by a number of appendices and a user's manual. These parts have the same number as this report, but are indexed B and C.

2. Symbols

This list of symbols contains the symbols used in the sections 1 to 12. The symbols that are used in the program text (see the appendices) are named in the program text itself, so they are omitted here.

A : Hinge
B : Hinge
O : Hinge
P : Hinge
OA : Anterior Cruciate Link (ACL)
AB : Femoral Link
OP : Tibial Link
PB : Posterior Cruciate Link (PCL)
a : Slope or concavity/convexity
a' : Vector component
A_i : Polynome coefficient
b' : Vector component
C_i : Length of vector
CA_x : Constant
CB_x : Constant
CC_x : Constant
h : Vertical displacement
IC : Instantaneous center of velocity
l : Distance centrode-contact profile
l_i : Length of link i
l_i'' : Distance A to IC
R_i : Rotation matrix
x_f : x-coordinate tibial centrode
x_t : x-coordinate tibial contact profile
x_0 : Horizontal displacement
y_f : y-coordinate tibial centrode
y_t : y-coordinate tibial contact profile

Greek symbols: angles (degrees)

α : Variable
α_s : α at input
β : Variable
γ : Flexion angle
δ : Variable
θ : $\angle OPB$
φ : $\angle AOP$
χ : Rotation angle

Vectors:

$\vec{A}$: Vector from O to A
$\vec{c}_{f,i}$: Tangent to femoral centrode
$\vec{c}_{t,i}$: Tangent to tibial centrode
$\vec{FC}$: From O to femoral centrode
$\vec{l}_i$: Vectors representing the four links
$\vec{n}_x$: Unit vectors
$\vec{r}_f$: From O to tibial centrode
$\vec{r}_t$: From O to tibial contact profile
$\vec{TC}$: From O to tibial centrode
$\vec{t}_t$: Tangent to tibial contact profile

3. Preface

In fundamental research parameter variation has become an often used tool. So it is necessary to give students the opportunity to train themselves in using the principle of parameter variation. The curriculum of the faculty of Medical Engineering Technology of the Eindhoven University of Technology therefore contains a simulation practical. In order to expand this practical a software program had to be designed, which would make a simulation with parameter variation possible. The object to be simulated was chosen to be a 2 dimensional kinematic model of the knee joint. The software should be able to calculate the effects of the change in various parameters, which further on in this paper will be specified.
This problem was considered very suitable to be solved in the form of one of the two larger practicals (about 240 hours each) that occur in our curriculum. This paper is the result of our work.
Because it would be foolish to do unnecessary work, we went searching for already existing solutions. There is a lot of literature on four bar mechanisms, but there seemed to be little interest in the type of solution we are looking for. Specialists on kinematics at the Eindhoven University of Technology made the same suggestion. Therefore we had to start from scratch.
When the mathematical model was finished, we expected that the programming should not take to much time, but here we were wrong. Especially the use of mathematical and graphical libraries proved to be cumbersome. After these problems had been tackled, the program worked satisfactorily, and we think it is a usable tool for a simulation practical. We hope it will be used for that purpose.
Finally we would like to thank the persons who have been very helpful to us. These persons are:

Prof. dr. A. Huson
Dr. ir. A.A.H.J. Sauren
Ing. H.A. Bulten
Drs. A.W.M. de Jong
Dr. G.J. Visser

Niek van Nunen
Frank Peters

4. Introduction

The bones that meet in a joint, as well as the ligaments that hold the bones together, can be analysed as a mechanical linkage. Since the movements allowed to the bones at the human knee occur mainly in the sagittal plane, much can be learned by treating the knee as a two-dimensional single-degree-of-freedom linkage moving in a single plane. *Figure 1* shows a human knee from which the lateral femoral condyle has been excised, exposing the cruciate ligaments. The ligaments, together with the two bones, form the cruciate linkage. *OP* will be called the ***tibial link***. *AB* is the ***femoral link***. The lines *OA* and *PB* represent the anterior (ACL) and posterior (PCL) cruciate ligaments, respectively.

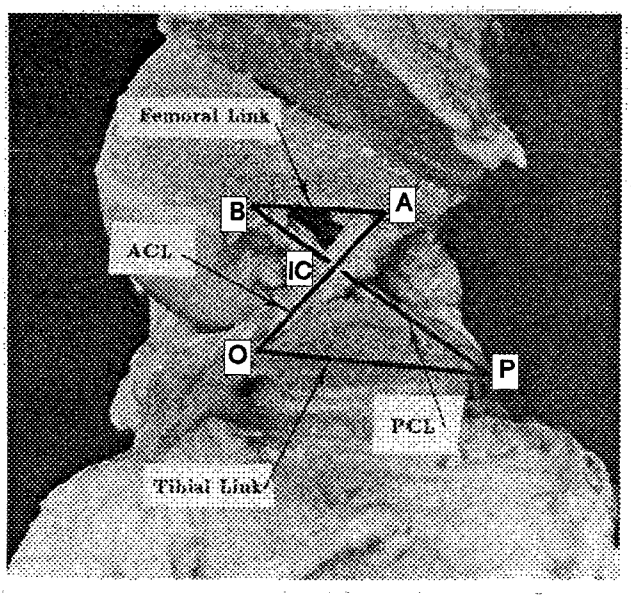


Figure 1.

As mentioned before, the initial task was to design or write a software-program which simulates a two-dimensional kinematic model of a knee joint. The program initially had to satisfy several demands. Whether or not it would be possilble to satisfy all demands would appear along the way. The initial task description was:

The model is based on the concept of a four bar mechanism. Two (not necessarily) crossed bars represent the cruciate ligaments. These two bars can be either rigid or slightly elastic. The other two bars each are expanded with a plane, in such a way that during the motion of the four bar mechanism there will be a continuous contact between the two profiles. These profiles represent the contact planes of the femur and tibia.

The input of the model contains:

- *the coordinates of the rotation centers of hinges of the links. Herewith the dimensions of the hinges are determined.*
- *The stiffness of the links.*
- *The profile of one of the two contact profiles.*
- *The initial position of the mechanism i.e. the flexion-angle of the joint.*
- *The increment of the flexion angle*

The output should contain:

- *The curve of the second profile.*
- *The position of the contact point.*
- *The position of the instantaneous center of rotation, as well as both centrodes.*
- *Instantaneous translation velocities (slip) in the contact point.*

The input possibilities of the parameter variation:

- *A new position for one or more of the revolutes of the bars.*
- *A change in the geometry of one of the contact profiles.*

The output of the parameter variation:

- *The strains in the bars in case of a change in the position of revolutes or contact profiles.*
- *the forces in the bars.*
- *forces in the contact point.*

In the next section a strategy will be presented, according to which the problem will be worked out.

5. Strategy

In order to carry on through the whole process smoothly it is necessary to use a well considered strategy. The strategy chosen is the following:

First the exact problem definition will be determined. Which of the above mentioned demands will be realised? Should all demands be realised, but then taking the risk not to be able to end up with a completely working software program (the time to be put in is about 240 hours)? The strategy chosen here is to first consider the most elementary parts of the problem, and then with an eye on the time left extend the model with the less important specifications. Whether or not any specifications will be implemented will be decided before starting to write the software program, so that the program can be structured well and does not have to be changed afterwards. What the most important specifications are will be determined in the next section.
The next step is the construction of a graphical analysis. This will give a first indication of how the mathematical problem is to be solved.
Then a mathematical model will be formulated which consists of both analytical and numerical parts. In this step the model will be formulated in such a way that the next step, the programming, can be carried out rather easily. Also an eye will be kept on the fact that the program shouldn't work to slow. The consequence of this is that sometimes numerical approximations will be necessary where an exact but slow solution is possible.
The last step contains the programming itself. This part also includes the testing of the program and if necessary the writing of a manual.

6. Adjusted problem definition

In this section a global view is presented from which specifications initially are intended to be worked out. Later on this may very well be extended or reduced depending on the time left.
In the first place the users of the software will have to be able to study the effects on the form of the femoral contact profile when the input consists of:

a. several geometries of the tibial contact profile.
b. crossed and uncrossed ligament configurations.
c. slip velocities.

These elements now will be worked out in the global method that will be used as the basis of the mathematical model.

7. Global method: a graphical analysis

7.1 The flexion angle γ

The basis of the model will be a simple four bar mechanism. This is one of the most elementary mechanisms in kinematics. For the mechanism configurations as drawn in *figure 2* are taken. To make a link to a real situation an arbitrary flexion angle γ is defined. *Figure 1* suggests that when the tibial link OP is parallel to the femoral link AB, the joint is flexed to about 50°. So positions can be defined where the joint is fully extended and fully flexed. From literature [1] it is known that when the angle α in *figure 2* is about 50°, the joint is in full extension (in other words, γ is zero). In other literature [2] it is found that the maximum flexion angle is about 140° (this means α is -90°). In the following analysis this range ($-90° < \alpha < 50°$) will be used as the range within which the calculations will be made.

7.2 The input.

When the software program is finished, the students will have to give some input. This input will consist of:

1. The form of the mechanism, i.e. whether the mechanism is crossed or uncrossed.
2. the coordinates of the hinges A and B. The coordinates of the hinges O and P are already determined. Herewith all the dimensions (i.e. lengths and a certain flexion angle γ) and kinematics are determined.
3. The curve of the tibial contact profile. The choice out of a straight line or a parabola will be given. The choice for a circle or an ellipse was neglected, because these curves are rather similar to the parabola, when the latter is only slightly concave or convex. These forms (straight line and parabola) can be adjusted by changing the parameters of the basic equation of each of the two possibilities.

When this input is given the situation is as shown in *figure 2*.
Now a method to calculate the curve of the femoral contact profile will be presented.

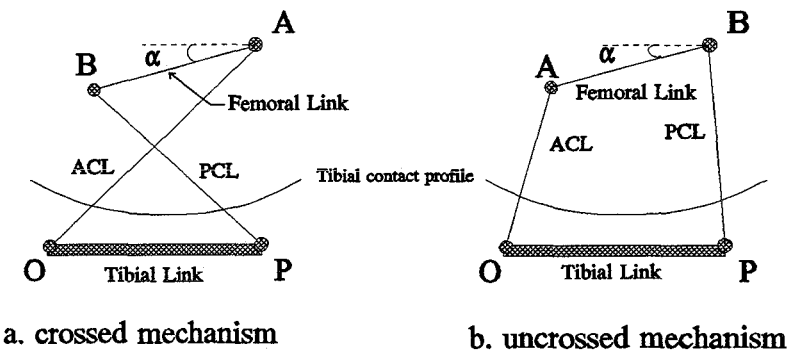


Figure 2. The two types of mechanisms

7.3 Construction of the centrodes.

When the relative positions of the links are changed, for example when we bend the knee, the tibial link and the femoral link carry out a motion relative to each other. At each position of the mechanism a point can be determined were the two links have zero relative velocity. This means that at this point the two links only rotate relative to each other. These points are called ***Instantaneous Centers of velocity (IC's)*** or ***poles***. In *figure 3* the IC is constructed for an uncrossed mechanism in a particular position.

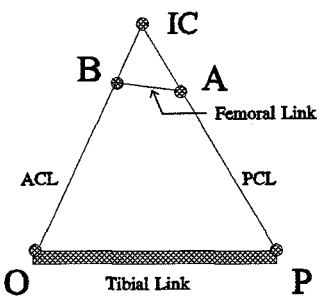


Figure 3. The Instantaneous Center of velocity (IC)

All IC's together (from all positions of the mechanism) form a curve, called a ***centrode.*** Two centrodes can be constructed. In the first case when the mechanism is moved with the tibial link fixed, which results in the ***tibial centrode*** and in the other case the femoral link is fixed, from which the ***femoral centrode*** is the result. These two centrodes are given in *figure 4*, for the crossed version of the mechanism.

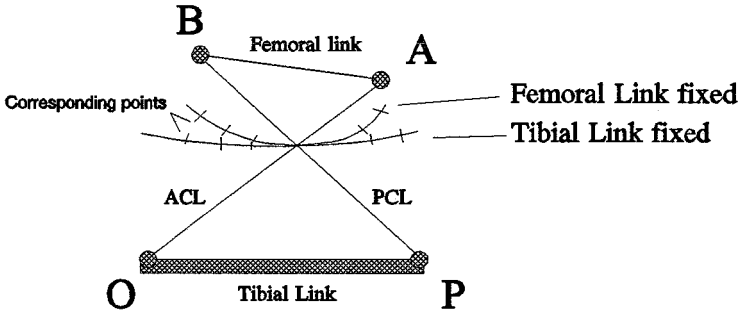


Figure 4. The corresponding points on the two centrodes.

It appears that these curves form a pure rolling contact. So along the curves corresponding points can be defined (one point of the tibial centrode will contact only one point of the femoral centrode). This is shown also in *figure 4*. With the help of these centrodes and the combinations of the corresponding points the curve of the femoral contact profile can be constructed.

7.4 Construction of the femoral contact profile.

From the given tibial profile a line perpendicular to this tibial profile is constructed (see *figure 5*). This line will intersect the tibial centrode, and will make an angle χ with the tangent to the centrode at the intersection point. The lenght of the line segment is l.
To maintain contact between the tibial and femoral contact profiles and to avoid pinching, the same combination of χ and l also has to be found on the femoral contact profile. Since it is known which point on the femoral centrode corresponds with the intersection point on the tibial centrode, a line with length l and an angle χ to the tangent of the femoral centrode can be drawn.

In this way a point of the femoral contact profile is identified (see *figure 5*). When the same procedure is followed for a sufficient number of points the whole femoral contact curve can be constructed.

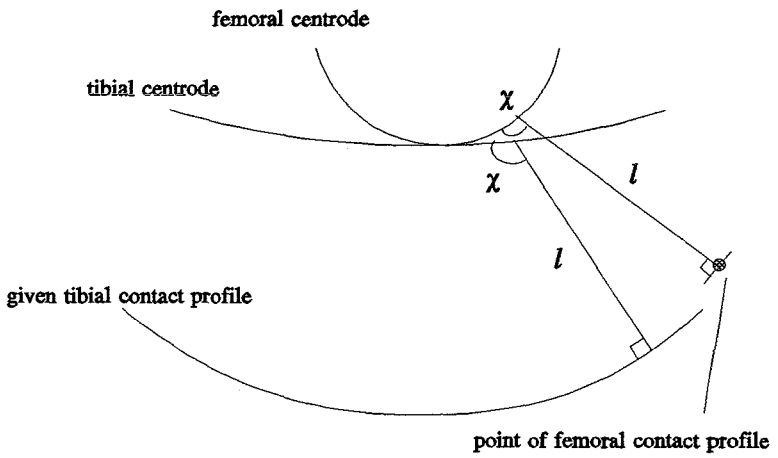


Figure 5. Construction of a point of the femoral contact profile.

This procedure will be translated into a mathematical model, which is needed to be able to write the software program.

8. Mathematical model

8.1 The centrodes

8.1.1 Definitions

In this analysis a planar four bar linkage *OABP* shall be considered (see *Figure 6*).

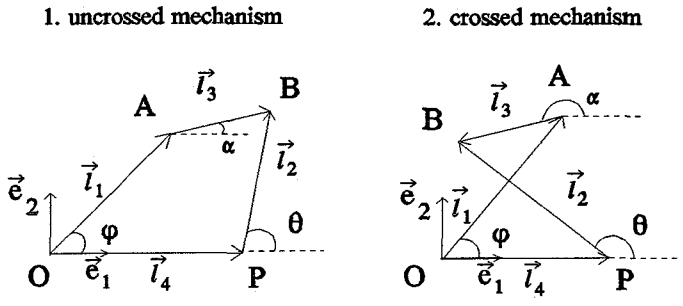


Figure 6. The two types of mechanisms.

l_1, l_2, l_3 and l_4 are the lenghts of the four links, and the angle α is the input. first the angles φ and θ will be calculated as a function of α. With the definitions as shown both the uncrossed and the crossed version can be described with the same formulae. The only difference will be the range of the angle α. In the uncrossed version α will vary between 50° and -90°, and in the crossed version the range will be from 230° to 90°. It is assumed here that for the chosen lenghts these two ranges are possible. Later on in the software program this will be checked and if necessary the range will be adapted. Now the following vectors are defined:

$$\vec{l}_1 = l_1\cos(\varphi)\vec{e}_1 + l_1\sin(\varphi)\vec{e}_2 \qquad (1)$$

$$\vec{l}_2 = l_2\cos(\theta)\vec{e}_1 + l_2\sin(\theta)\vec{e}_2 \qquad (2)$$

$$\vec{l}_3 = l_3\cos(\alpha)\vec{e}_1 + l_3\sin(\alpha)\vec{e}_2 \qquad (3)$$

$$\vec{l}_4 = l_4 \vec{e}_1 \tag{4}$$

Figure 6 shows that the following equation is valid in any position:

$$\vec{l}_1 + \vec{l}_3 = \vec{l}_2 + \vec{l}_4 \tag{5}$$

Using (5) $\varphi(\alpha)$ and $\theta(\alpha)$ can be calculated. $\varphi(\alpha)$ will be calculated first:

8.1.1.1 The angle $\varphi(\alpha)$

The lengths of the links are fixed, so

$$(\vec{l}_1 + \vec{l}_3 - \vec{l}_4) \cdot (\vec{l}_1 + \vec{l}_3 - \vec{l}_4) = l_2^2 \tag{6}$$

Substitution of (1),(3) and (4) then yields:

$$(l_1 \cos\varphi + l_3 \cos(\alpha) - l_4)^2 + (l_1 \sin\varphi + l_3 \sin\alpha)^2 = l_2^2 \tag{7}$$

After some simplifications this can be written as:

$$CA_\varphi \sin\varphi + CB_\varphi \cos\varphi = CC_\varphi \tag{8}$$

where

$$CA_\varphi = \sin\alpha \qquad CB_\varphi = \cos\alpha - \frac{l_4}{l_3} \qquad CC_\varphi = \frac{l_4}{l_1}\cos\alpha + \frac{-l_1^2 + l_2^2 - l_3^2 - l_4^2}{2 l_1 l_3} \tag{9}$$

Now equation (8) has to be solved. Such a solution is obtained by expressing sin φ and cos φ in terms of $\tan(\varphi/2)$,

$$\sin\varphi = \frac{2\tan(\frac{\varphi}{2})}{1+\tan^2(\frac{\varphi}{2})} \qquad \cos\varphi = \frac{1-\tan^2(\frac{\varphi}{2})}{1+\tan^2(\frac{\varphi}{2})} \tag{10}$$

Substitution and simplification then yields:

$$\tan(\frac{\varphi}{2}) = \frac{CA_\varphi \pm \sqrt{CA_\varphi^2 + CB_\varphi^2 - CC_\varphi^2}}{CB_\varphi + CC_\varphi} \tag{11}$$

So two distinct values of φ are found:

$$\varphi^{+}(\alpha)=2\arctan\frac{CA_{\varphi}+\sqrt{CA_{\varphi}^2+CB_{\varphi}^2-CC_{\varphi}^2}}{CB_{\varphi}+CC_{\varphi}} \tag{12}$$

$$\varphi^{-}(\alpha)=2\arctan\frac{CA_{\varphi}-\sqrt{CA_{\varphi}^2+CB_{\varphi}^2-CC_{\varphi}^2}}{CB_{\varphi}+CC_{\varphi}} \tag{13}$$

In section 8.1.2 these results will be discussed.

8.1.1.2 The angle $\theta(\alpha)$

For the calculation of $\theta(\alpha)$ the following equation can be used:

$$(\vec{l}_2-\vec{l}_3+\vec{l}_4)\cdot(\vec{l}_2-\vec{l}_3+\vec{l}_4)=l_1^2 \tag{14}$$

When the same procedure as in (7) until (13) is followed, this results in:

$$\theta^{+}(\alpha)=2\arctan\frac{CA_{\theta}+\sqrt{CA_{\theta}^2+CB_{\theta}^2-CC_{\theta}^2}}{CB_{\theta}+CC_{\theta}} \tag{15}$$

$$\theta^{-}(\alpha)=2\arctan\frac{CA_{\theta}-\sqrt{CA_{\theta}^2+CB_{\theta}^2-CC_{\theta}^2}}{CB_{\theta}+CC_{\theta}} \tag{16}$$

where

$$CA_{\theta}=\sin\alpha \qquad CB_{\theta}=\cos\alpha-\frac{l_4}{l_3} \qquad CC_{\theta}=-\frac{l_4}{l_2}\cos\alpha-\frac{l_1^2-l_2^2-l_3^2-l_4^2}{2l_2l_3} \tag{17}$$

How these results can be interpreted and how they can be used, will be explained in the next section.

8.1.2 Interpretation of results

As shown in the previous section, there are, given an angle α, two solutions for φ and θ. What this means for the possible forms of the mechanisms is shown in *figure 7*. With the help of this figure it is easy to see that the solutions which have any fysical meaning are $\varphi^-(\alpha)$ and $\theta^-(\alpha)$. These two solutions are the positive angles, since in both formulae (12) and (13) as in formulae (15) and (16) the terms $CB + CC$ appear to be negative. So these two solutions will be used, and α will vary between 50° and -90° in case of the uncrossed mechanism, and between 230° and 90° in case of the crossed mechanism,as already mentioned in section 7.1.

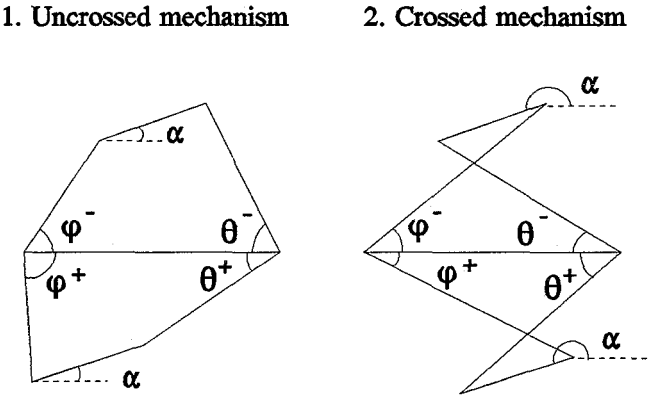


Figure 7. The possible solutions for z and h.

8.1.3 The tibial centrode

In order to find the position of the IC of the tibial and femoral links it is necessary to calculate the intersection point of the vectors l_1 and l_2 (the ligaments). The unit vectors of l_1 and l_2 need to be determined for this purpose.

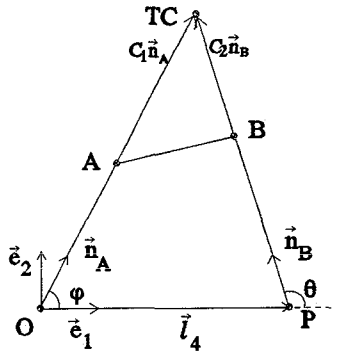


Figure 8. Construction of a point of the tibial centrode.

This results in (see also *figure 8*):

$$\vec{n}_A=\cos(\varphi)\vec{e}_1+\sin(\varphi)\vec{e}_2 \qquad \vec{n}_B=\cos(\theta)\vec{e}_1+\sin(\theta)\vec{e}_2 \qquad (18)$$

The intersection point now is defined by:

$$C_1\vec{n}_A=\vec{l}_4+C_2\vec{n}_B \qquad (19)$$

To find the coordinates of the intersection point it is sufficient to calculate only C_1. After substitution of (18) in (19) and some simplifications, this results in:

$$C_1=\frac{l_4\sin(\theta)}{\sin(\theta-\varphi)} \qquad (20)$$

so

$$\vec{TC}=\frac{l_4\sin(\theta)}{\sin(\theta-\varphi)}(\cos(\varphi)\vec{e}_1+\sin(\varphi)\vec{e}_2) \qquad (21)$$

Hereby two assumptions are made. The first one is that φ and θ will not become equal to 0° or 180°. The second assumption is that φ and θ do not become equal to each other. In case of realistic mechanisms these assumptions are fair.
Now the femoral centrode will be calculated, using the found tibial centrode. Here again the calculations are made for the uncrossed mechanism, and the formulae which will be found are valid also for the crossed mechanism (again only the range is rotated over 180°).

8.1.4 The femoral centrode

The components of the vector l_1" with length l_1", which points from A to FC (see *Figure 9c*), now have to be calculated. These components can be calculated as follows:

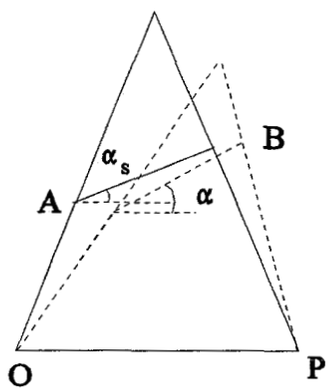


Figure 9a.

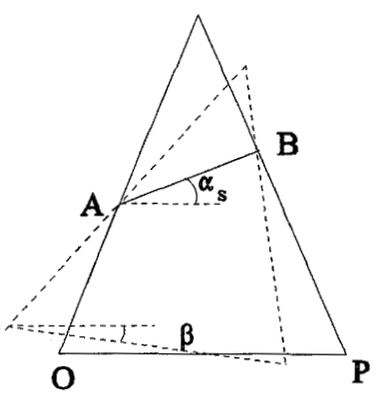


Figure 9b.

When the tibial centrode was calculated, the tibial link was fixed. From this position the angle α was changed and the angles φ and θ were calculated. Now the femoral centrode is calculated by fixing the femoral centrode. The angle α in the input situation now will be called α_s. When the mechanism is moved into different positions with the tibial link fixed, an angle α different from α_s can be found (see *Figure 9a)*. But since the femoral link was to be fixed, this position has to be translated and rotated so, that the tibial link is again in the input situation (see *Figure 9b*). In this way an angle β is found.

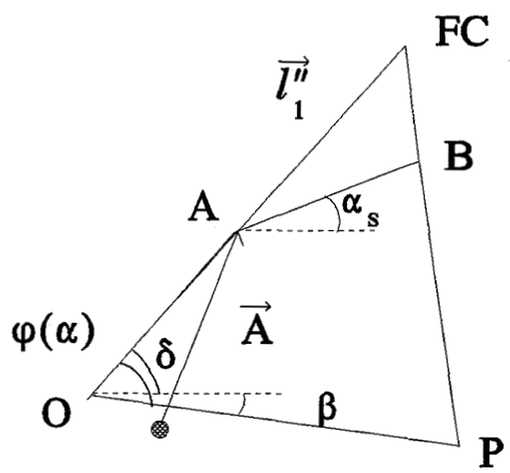


Figure 9c: Construction of a point of the femoral centrode.

Now it is clear that:

$$\beta = \alpha - \alpha_s \tag{22}$$

Now angle δ is defined by:

$$\delta = \varphi - \beta = \varphi - \alpha + \alpha_s \tag{23}$$

Furthermore l_1" is determined by:

$$l_1'' = |l_1 - C_1| \tag{24}$$

The position of FC now can be calculated with:

$$\vec{FC} = \vec{A} + a'\vec{e}_1 + b'\vec{e}_2 \tag{25}$$

Where:

$$a' = l_1'' \cos(\delta + R) \qquad b' = l_1'' \sin(\delta + R) \tag{26}$$

Note that in this last formula R is equal to zero in case of the uncrossed mechanism and equal to π in case of the crossed mechanism!!
Now that both the tibial and femoral centrode are known, the femoral contact profile can be constructed. This will be worked out in section 8.2.

8.2 The femoral contact profile

8.2.1 The distance l

Since the two centrodes have been constructed in the previous sections, the curve of the femoral contact profile now also can be constructed. To this aim the combination (α,l) has to be found, according to the description given in section 7. The length l is calculated by using the next procedure (see also *figure 10*):

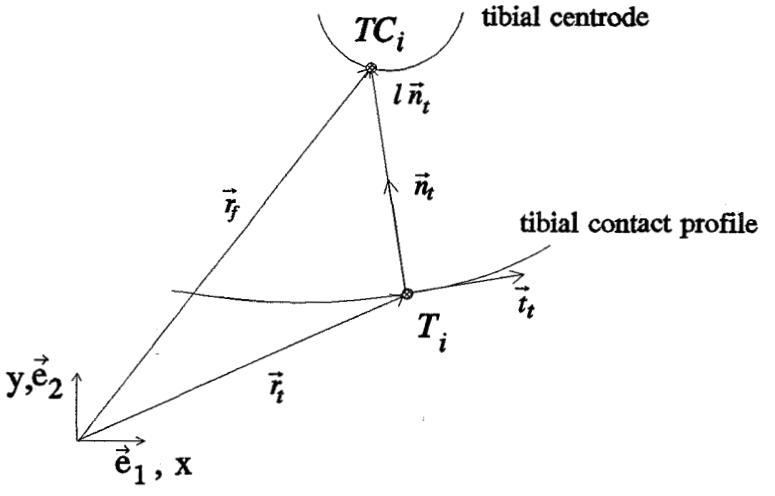


Figure 10. construction of unit vector n_t and distance l

Given the profile in *figure 10*, the position vector r_t is defined by:

$$\vec{r}_t = x\vec{e}_1 + y(x)\vec{e}_2 \tag{27}$$

Differentiation of the position vector r_t with respect to x yields:

$$\frac{d\vec{r}_t}{dx} = \vec{e}_1 + y'\vec{e}_2 \tag{28}$$

t_t is the unit tangent vector to the tibial contact profile at position r_t, so:

$$\vec{t}_t(x) = \frac{d\vec{r}_t}{dx} \Big/ \left\| \frac{d\vec{r}_t}{dx} \right\| = \frac{1}{\sqrt{1+y_t'^2}}(\vec{e}_1 + y_t'\vec{e}_2) \tag{29}$$

The vector n_t normal to t_t now can be found by:

$$\vec{n}_t(x)=\frac{d\vec{t}_t}{dx}/\left\|\frac{d\vec{t}_t}{dx}\right\| \qquad ;|\vec{n}_t(x)|=1 \tag{30}$$

Working this out yields:

$$\vec{n}_t(x)=\frac{1}{\sqrt{1+y_t'^2}}(-y_t'\vec{e}_1+\vec{e}_2) \tag{31}$$

Since the tibial centrode already is calculated, the vector r_f is known. Now vector r_t and l have to be found, such that:

$$\vec{r}_t+l\vec{n}_t=\vec{r}_f \tag{32}$$

where:

$$\vec{r}_t=x_t\vec{e}_1+y_t(x_t)\vec{e}_2 \tag{33}$$

and:

$$\vec{r}_f=x_f\vec{e}_1+y_f\vec{e}_2 \tag{34}$$

Substitution of (31), (33) and (34) in (32) yields:

$$x_t\vec{e}_1+y_t(x_t)\vec{e}_2+\frac{l}{\sqrt{1+y_t'^2}}(-y_t'\vec{e}_1+\vec{e}_2)=x_f\vec{e}_1+y_f\vec{e}_2 \tag{35}$$

Now (35) can be divided into two components:

$$\vec{e}_1: \quad x_t-\frac{ly_t'}{\sqrt{1+y_t'^2}}=x_f \tag{36}$$

$$\vec{e}_2: \quad y_t(x_t)+\frac{l}{\sqrt{1+y_t'^2}}=y_f \tag{37}$$

In these equations y_t' can be expressed in terms of x_t. Herewith there are two equations with two unknowns (x_t and l). So the variable l can be found. These results now will be used for different types of tibial contact profiles.

8.2.1.1 Tibial contact profile is a straight line:

A straight line is described as in the next equation:

$$y_t=ax_t+h \tag{38}$$

Differentiation of (38) with respect to x yields:

$$y_t'=a \tag{39}$$

Substitution of (38) and (39) in (36) and (37) results in:

$$x_t-\frac{la}{\sqrt{1+a^2}}=x_f \tag{40}$$

and

$$ax_t+h+\frac{l}{\sqrt{1+a^2}}=y_f \tag{41}$$

The next step is to solve l and x_t from (40) and (41). This yields:

$$x_t=\frac{ay_f-ah+x_f}{1+a^2} \qquad l=\frac{y_f-ax_f-h}{\sqrt{1+a^2}} \tag{42}$$

A special case is a straight horizontal line ($a=0$). Then the results are:

$$x_t=x_f \qquad l=y_f-h \tag{43}$$

8.2.1.2 Tibial contact profile is a parabola:

A parabola is a second order polynomial and can be described by:

$$y_t=a(x_t-x_0)^2+h \tag{44}$$

$(x_0\ ,\ h)$ is the displacement from the origin $O(0,0)$, and a determines whether the curve is convex or concave, and also the convexity or concavity. Differentiation of (44) with respect to x yields:

$$y_t'=2a(x_t-x_0) \tag{45}$$

In order to be able to solve this problem, first l has to be eliminated from (38) and (39). This yields:

$$x_t-y_t'(y_f-y_t(x_t))=x_f \tag{46}$$

Substitution of (44) and (45) in (46) yields:

$$2a^2x_t^3-6a^2x_0x_t^2+(1-2ay_f+2ah+6a^2x_0^2)x_t-2ahx_0+2ax_0y_f-2a^2x_0^3-x_f=0 \tag{47}$$

The same equation can be represented as:

$$A_0x_t^3+A_1x_t^2+A_2x_t+A_3=0 \tag{48}$$

with:

$$A_3=2ax_0y_f-2ahx_0-2a^2x_0^3-x_f \qquad A_2=1-2ay_f+2ah+6a^2x_0^2 \tag{49}$$

$$A_1=-6a^2x_0 \qquad A_0=2a^2 \tag{50}$$

This is a cubic equation which has to be solved numerically.
How this will be done, will be dicussed in section 9.3.3. Once x_t is known, l can be calculated with:

$$l=(y_f-a(x_t-x_0)^2-h)\sqrt{4a^2(x_t-x_0)^2+1} \tag{51}$$

8.2.2 The angle χ_i

At this point of the mathematical analysis it is necessary to switch over to a discretized formulation. So from here on we will also use the index *i* at some of the variables. These variables however are subject to the same relations as before!!
Before χ_i , which is the angle between the vector $n_{t,i}$ and the tangent of the tibial centrode, can be calculated, the vector $c_{t,i}$ tangent to the tibial centrode has to be found. It is possible to find an analytical solution for this problem, but this solution will be very complex and will take a lot of calculation time. So a rather simple but satisfactory approximation is chosen (see also figure 11):

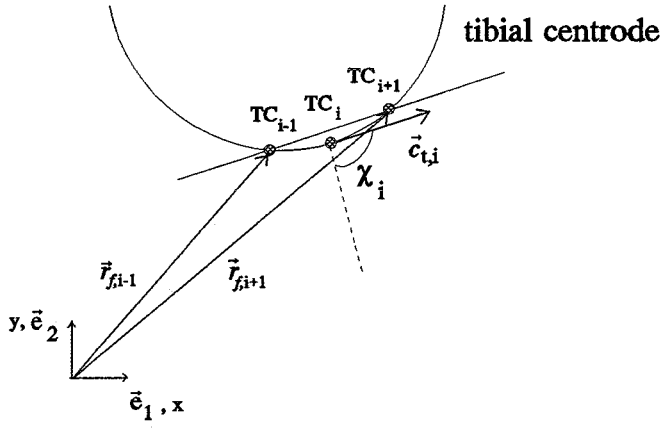


Figure 11. Approximation of vector $c_{t,i}$ tangent to tibial centrode.

A number of points on the tibial centrode are known, and since the curve from the tibial centrode is rather smooth, a good approximation is found by using the tangent of the line through the point TC_{i-1} and the point TC_{i+1}. This results in:

$$\vec{c}_{t,i}=\vec{TC}_{f,i+1}-\vec{TC}_{f,i-1}=(TC_{x,i+1}-TC_{x,i-1})\vec{e}_1+(TC_{y,i+1}-TC_{y,i-1})\vec{e}_2 \qquad (52)$$

Now χ_i is found by taking the dot product of (28) and (48):

$$\vec{c}_{t,i}\cdot\vec{n}_{t,i}=|\vec{c}_{t,i}|\,|\vec{n}_{t,i}|\cos(\pi-\chi_i) \qquad (53)$$

This results in:

$$\chi_i = \pi - \arccos \frac{(\vec{n}_{t,i} \cdot \vec{c}_{t,i})}{|\vec{n}_{t,i}| \, |\vec{c}_{t,i}|} \tag{54}$$

8.2.3 The construction of the femoral contact profile

Before the femoral profile can be constructed, an approximation of the unit vector tangent to the femoral centrode, $c_{f,i}$, has to be made. Hereto the same approximation as in equation (52) is used. This results in:

$$\vec{c}_{f,i} = \frac{1}{S_i}(\vec{FC}_{i+1} - \vec{FC}_{i-1}) = \frac{1}{S_i}((FC_{x,i+1} - FC_{x,i-1})\vec{e}_1 + (FC_{y,i+1} - FC_{y,i-1})\vec{e}_2); \; |\vec{c}_{f,i}| = 1 \tag{55}$$

Where:

$$S_i = \sqrt{(FC_{x,i+1} - FC_{x,i-1})^2 + (FC_{y,i+1} - FC_{y,i-1})^2} \tag{56}$$

A point F_i from the femoral contact profile now can be found by rotating the unit vector $c_{f,i}$ over an angle χ_i, multiplication of the found vector with l, and finally addition of vector FC_i. The rotation is done by multiplying the vector with a rotation matrix R_i. See *figure 12.*

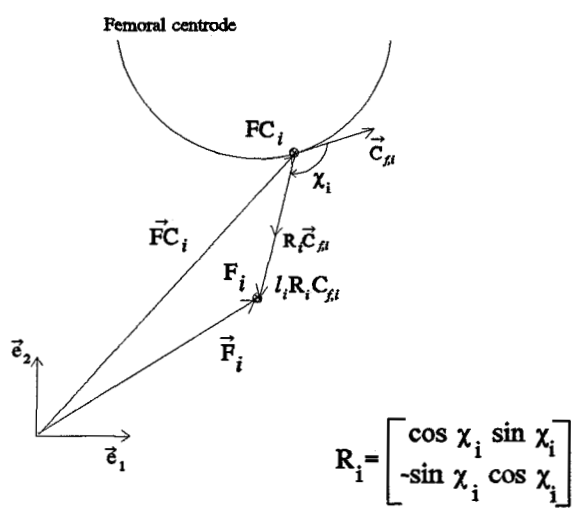


Figure 12. Determination of a point of the femoral contact profile by rotation and multiplication.

This then results in:

$$\vec{F}_i = \vec{FC}_i + l_i \cdot R_i \cdot \vec{c}_{f,i} = F_{x,i}\vec{e}_1 + F_{y,i}\vec{e}_2 \qquad (57)$$

When this procedure of calculating the coordinates of a point of the femoral contact profile is done for a sufficient number of points, a good indication of the contact profile is given by drawing straight lines between the points. Curve fitting is also possible, but may cost a lot of calculation time. This "visualisation of results" is one of the aspects, that will be discussed in the following section.

9. The programming

9.1 Program structure

Using the mathematical model a software program has been written. The program consists of six main parts. These are:

1. Input
2. Calculation
3. Output
4. Parameter variation
5. Slip ratio
6. Movie

Each part of this structure is written as a subroutine, which is called from the main program. All subroutines may use other subroutines, for example from a library. *Figure 13* shows this.

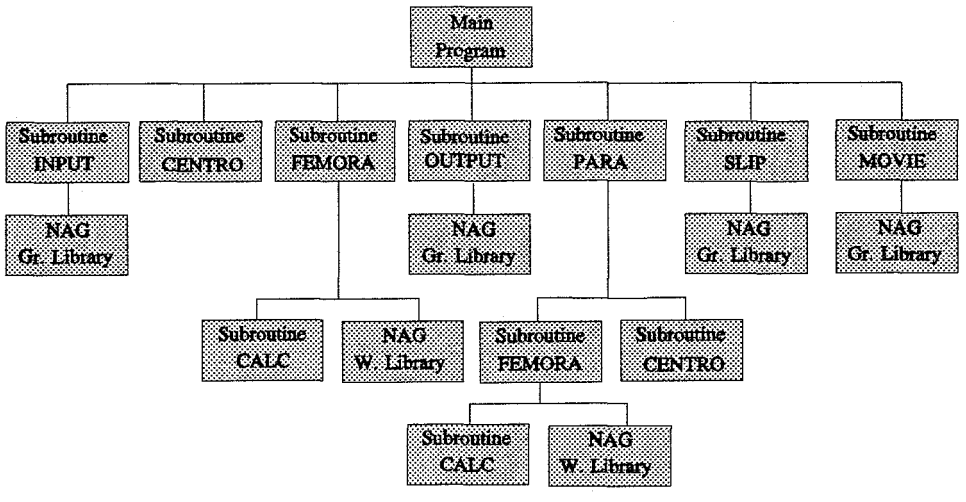


Figure 13. The structure of the software program: main program and subroutines. For further information on the software used, see Chapter 12, Literature.

Now the routines used will be specified. It is recommended here to keep the program text (see *Appendices A to I*) aside. This is because the program text contains a lot of comment lines, which can be very useful in understanding the structure of the program.

9.2 The main program

The main program calls the subroutines neede to perform the operations requested by the user. These operations are:

1. Input of a mechanism (after input the centrodes and contact profiles are calculated automatically).
2. Plotting of results.
3. Parameter variation.
4. Slip ratio.
5. Movie.
6. Quit the program.

9.3 The subroutines

As mentioned, the program consists of a number of subroutines. Each of them performs a specified task, and is called from the main program. In this way the program is well structured, and easy to understand.
From each subroutine now a short description will be given.

9.3.1 Subroutine INPUT

This subroutine supplies the program with the input parameters. These are:

1. Is the mechanism crossed or uncrossed?
2. The coordinates of A and B.
3. Type of tibial contact profile (straight line or parabola).
4. Parameters for the tibial contact profile.
5. The number of increments.
6. The choice for the dimensions of a perspex model which is available (no further input has to be given). This model will also be used for the validation of the program.
7. The choice for realistic anatomic dimensions which will be specified later on in the user's manual.

9.3.2 Subroutine CENTRO

In this subroutine the tibial and the femoral centrode are calculated. First the lenghts of the links have to be calculated. This is done by:

$$l_1=\sqrt{x_A^2+y_A^2}$$

$$l_2=\sqrt{(x_B-x_P)^2+y_B^2}$$

$$l_3=\sqrt{(x_B-x_A)^2+(y_B-y_A)^2}$$

$$l_4=x_P$$

Now the lenghts of the links are known, φ and θ as a function of α can be calculated, by using equations (12) and (15) (section 8.1.1).
As mentioned before, for α a range from 50° to -90° (for the uncrossed version; for the crossed version the range will be 230° to 90°) will be chosen. This range has to be divided into the chosen number of increments. This is done with the following equation:

$$\alpha=\frac{50\pi}{180}+R-\frac{I-1}{N-1}\frac{140\pi}{180}$$

In this equation R is equal to zero if the mechanism is an uncrossed one, and equal to π when the mechanism is a crossed one. I is the I^{th} increments and N is the total number of increments.
Next the vector *TC*, which represents the tibial centrode and vector *FC*, which represents the femoral centrode, are determined. It may very well be possible that the mechanism cannot make the full range of 140° of flexion. This is caused by the fact that the mechanism may change from the crossed version into the uncrossed version or the other way around. Then a so called **Tail** appears (see *Figure 14*). The curves then will be of such a form that moving the mechanism becomes impossible. When this happens the program will give a warning, and it will also give the range that can be covered.

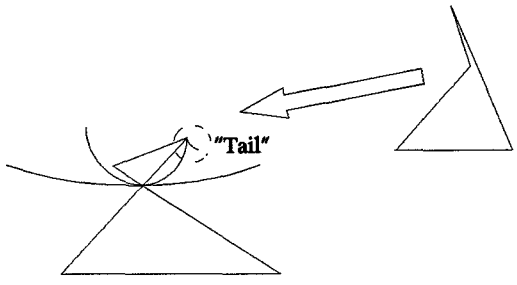


Figure 14: The problem that appears when the mechanism changes from the crossed to the uncrossed version. The program will detect the beginning of the "Tail" and stop further calculations of the centrode.

9.3.3 Subroutine FEMORA

This subroutine calculates the femoral contact profile. It consists of two parts. One of them calculates the femoral contact profile in case the tibial contact profile is a straight line and the other in case the tibial contact profile is a parabole:
when the tibial contact profile is a straight line the mathematical analysis (section 8) together with the program text (see Appendices) give enough information. When the tibial contact profile is a parabola however, some more explanation is necessary.
In section 8.2.1.2 it was found that a cubic equation had to be solved. To be able to solve this equation a routine from the NAG-library is used. This routine is called C02AGF. This routine will give three solutions. For this problem only one of these solutions is of any interest. This particular solution has to be selected.
In the first place it is checked whether or not there are any complex solutions. If so, there is only one real solution, and this has to be the right solution. But when there are three real solutions (two real solutions is inpossible since the complex solutions always appear in conjugates) the right solution has to be selected. The right solution then is the one which is nearest to the previous solution. This means that for the first point already a solution has to be known. It is known that when $x = x_0$ the line searched is a vertical one. so then the solution of the polynome also has to be x_0. In the program that point of the array of points of the tibial centrode is selected from which the x-coordinate is the closest to x_0. For this "starting point" the polynome is solved, and the solution which is nearest to x_0 is chosen. For the next points the same procedure can be repeated. The consequence of this method is that the femoral contact profile has to be calculated in two parts. One part is calculated from the starting point back-

wards until $I=1$, and the other part is calculated from the starting point forwards, until $I=N$.
The solution however gives some problems. When a mechanism is chosen it may happen that there are points with one real solution for the cubic equation, and also points with three real solutions. The method used here then may have some difficulties in finding the right solution. Here also a warning is given. This effect can be eliminated by changing either the convexity/concavity a of the parabola (the parabola will have to be flatter, this means a has to be closer to zero) or the vertical displacement h (h will have to be larger).
This whole solution of the cubic equation has been worked out in the subroutine CALC, which is not mentioned separately, since it is only a part of subroutine FEMORA.
Subroutine FEMORA now will calculate a point of the femoral contact profile by rotation and multiplication of the vector tangent to the femoral centrode.

9.3.4 Subroutine OUTPUT

This subroutine visualises the results from the calculations. It will draw the mechanism, according to the given coordinates of points A and B. Next it will draw the centrodes and contact profiles. In case of parameter variation the centrodes and contact profiles are drawn three times, once for the initial parameter values, once with the chosen parameter larger and once with the chosen parameter smaller (values are mentioned in the user's manual). The three cases will be drawn in different linestyles. The curves consist of straight lines between the calculated points. Curve fitting was not chosen, because this would take a lot of extra calculation time. Furthermore, the program will show that when a sufficient number of increments is chosen, the curves appear to be smooth enough.

9.3.5 Subroutine PARA

In this last part of the program the opportunity is given to change some variables. The following changes are possible:

1. A change in one of the coordinates of points A and B (x- or y-coordinate). The program then will calculate the effects on the femoral contact profile when the chosen coordinate is made a certain value smaller and larger. This value is mentioned in the user's manual.

2. A change in the curve of the tibial contact profile. The effects of a little more and a little less concave or convex curve on the femoral contact profile will be shown. A vertical or horizontal translation of the curve also is possible (again see the user's manual for the exact values).
3. A third possibility is to make one or more ligaments slightly elastic. At this point it has to be concluded that there is no time left for this part of the problem.

After the parameter to be varied has been chosen, all calculations are done twice again, once with the chosen parameter is a certain value smaller and once with the chosen parameter is a certain value larger (see user's manual for exact values).

9.3.6 Subroutine SLIP

This subroutine calculates the slip ratio, which is defined as the ratio of the lenght-segments of the two contact profiles over an interval of the flexion angle γ.
The slip ratio can be defined in a more mathematical way as follows:
A segment of the tibial contact profile is defined by:

$$dl_{tibial}=\sqrt{(T_{x,i+1}-T_{x,i})^2+(T_{y,i+1}-T_{y,i})^2}$$

The length of the corresponding segment of the femoral contact profile is defined by:

$$dl_{femoral}=\sqrt{(F_{x,i+1}-F_{x,i})^2+(F_{y,i+1}-F_{y,i})^2}$$

The slip ratio is now calculated by:

$$Slip=\frac{dl_{femoral}}{dl_{tibial}}$$

Also the average slip ratio is determined by calculating the ratio of the total lenghts of both the contact profiles. The total lengths of the profiles are calculated by summation of dl_{tibial} and $dl_{femoral}$.
The results of these calculations are represented in a diagram where the ratio is set out against the flexion angle γ.

9.3.7 Subroutine MOVIE

This last subroutine plots the mechanism and the four curves in several different positions. This routine gives a clear view of the whole process. It will play an important role in the model validation that is described in the next section.

10. Model validation

After the program text was entered and compiled (for the used equipment and software, see Lit. [6] and [7]) the program works satisfactorily. This fact however does not guarantee that the results are valid. Therefore some tests are performed. These tests will be described in this section.

10.1 Conditions resulting from the model

There must be one contact point between the tibial centrode and the femoral centrode. This contact point has to be at the intersection point of the Anterior Cruciate Ligament (ACL) and the Posterior Cruciate Ligament (PCL)of the mechanism in the position as given in the input. In the same way there must be one contact point between the two contact profiles. The tangent line of the contact point of the contact profiles has to intersect the contact point of the centrodes.
When the tibial contact profile is chosen as a straight line, it is easy to approximate the angle between the vector n_t and the tibial centrode. This angle has to be found again at the femoral centrode and the femoral contact profile.
When the tibial contact profile is a horizontal line, the vector n_t only has a vertical component, so each point of the femoral contact profile lies straight above the tibial contact profile. (In practice these last two conditions only can be validated for the first and the last point of the curves).

10.2 Symmetrical Mechanism

When a symmetrical mechanism is chosen, the calculated centrodes have to be symmetrical as well. In the same way the calculated femoral contact profile has to be symmetrical as the chosen tibial contact profile (and the mechanism) is symmetrical. Symmetrical in this case means symmetrical about the line $x = 50$ (the x- coordinate of P was chosen to be 100).

10.3 The Perspex Model

In this case the dimensions of a realisation in perspex of the model were used as input parameters for the program. This perspex exhibited all the kinematical characteristics belonging to its particular configuration. The program calculated the tibial and femoral centrodes and the femoral

contact profile, which have to be in agreement with the perspex model.

10.4 Subroutine Movie

The most powerful possibility to validate the model however is the in section 9.3.6. named subroutine MOVIE. When the mechanism is drawn in several positions, a smooth movement of the curves should appear, where the curves form a rolling contact.

When the tests mentioned are performed on several mechanisms, it can be concluded that the model programmed is correct.

11. Conclusions and recommendations

The following conclusions and recommendations can be made.

1. The model used in the program calculates the correct centrodes and contact profile.
2. It appeared to be impossible to satisfy all the demands given in the problem definition in section 4. The reason for this is that there was to little time. The program calculates the centrodes and the femoral contact profile, and it is also possible to calculate the effects of a variation in one of the parameters.
3. The program certainly is not written in the most efficient way. First many comment lines were used to make the program text better understandable. Second some arrays are larger than necessary, so the program will use a rather large amount of memory. Third the input needs to be correct. When, for example a string is entered where a number is expected, the program is terminated, instead of asking the same question again. Finally not all the equations are represented in the most accurate algebraic form. This however is of no significant meaning, since the calculations are accurate enough for the purpose which has to be served.
   All these restrictions are due to a lack of programming experience and knowledge.
4. The program runs faster than expected on a 80386 PC/AT with a 80387 coprocessor. The most complex calculations only take a few seconds, even when the maximum number of increments is chosen. Tests on other PC's showed that a 20286 with coprocessor is sufficient also, but a PC/XT is not usable, since it is too slow. Because the speed is high enough, one may consider to calculate the exact tangents of the centrodes, where now an approximation is used. Validation of the model however showed that the results need not be more accurate than they already are.
5. The program is easy to use, because of the shell that is built around the calculations.
6. When a mechanism is used which has realistic dimensions it appears that when the links are rigid it is not possible to reach 140° flexion. This may be an indication that both the links and the contactprofiles are in fact not rigid.
7. Instead of using a manual it may be useful to implement a help function which can be called at any place in the program.

8. The use of defaults may be of some help, when only some small changes are to be entered.
9. This report does not mention the worth of this program. When used in a practical some background information is needed for a better understanding.
10. The program can be extended easily because of the structure of a main program and subroutines.
11. It may be possible that some of the problems mentioned will be worked out after this report is completed. These changes may cause some difference in the program text (see Appendices). These changes will appear mainly in the main program, since extra functions will be written in the form of subroutines that are called from the main program.

Some of the problems that were mentioned in these conclusions may be joined in a new practical. This practical may also include the parts we were not able to solve. In particular the part where the links are not rigid anymore seems interesting. This problem however demands a different, somewhat more mechanical approach.

12. Literature

[1] Ian A.F. Stokes (Editor)
Mechanical factors and the skeleton
John Libbey & Company Limited, London
ISBN 0-86196-006-8

[2] Dale Daniel, Wayne Akeson, John O' Connor
Knee Ligaments; Structure, Function, Injury and Repair
Raven Press, New York
ISBN 0-88167-605-5

[3] Richard S. Hartenberg and Jacques Denavit
Kinematic Synthesis of Linkages
McGraw-Hill series in MECHANICAL ENGINEERING
McGraw-Hill Book company, 1964

[4] dr. G.R. Veldkamp
Kinematica
Scheltema & Holkema, Amsterdam, 1970
ISBN 90-6060-502-0

[5] A. Huson, C.W. Spoor and A.J. Verbout
A Model of the Human Knee, Derived from Kinematic Principles and Its Relevance for Endoprothesis Design.
Acta Morphologica Neerlando-Scandinavica 27 (1989), pp. 45-62

[6] Software used:

Wordperfect 5.1 NL
Drawperfect 1.1 NL
MS-Fortran 5.0 Compiler
NAG Workstation Library
NAG PC Graphics library
Norton Classic Editor, version 1.5

All software used was retrieved from the Calculation Center (Rekencentrum) of the Eindhoven University of Technology.

[7] Hardware used:

Rembrant 386 AT with 80387 coprocessor
VGA card with colour monitor

This hardware was retrieved from the Department of Fundamental Research, Faculty of Mechancal Engineering of the Eindhoven University of Technology.

Appendices

Report number 92.013 B

Contents

Contents 1
Appendix A: Main Program 2
Appendix B: Subroutine INPUT 9
Appendic C: Subroutine CENTRO 14
Appendix D: Subroutine FEMORA 18
Appendix E: Subroutine OUTPUT 24
Appendix F: Subroutine PARA 27
Appendix G: Subroutine SLIP 32
Appendix H: Subroutine MOVIE 35

Appendix A: Main Program

```
C     ***********************************************
C     ** THIS PROGRAM IS WRITTEN BY NIEK VAN NUNEN **
C     **               AND FRANK PETERS              **
C     ***********************************************
C
      PROGRAM KNEE
      IMPLICIT NONE
C
C     ***********************************************************
C     ** THIS IS THE MAIN PROGRAM FROM WHICH THE NECESSARY **
C     **           SUBROUTINES WILL BE CALLED.                 **
C     ***********************************************************
C
C     ************************************
C     ** DECLARATION OF THE IDENTIFIERS **
C     ************************************
C
      INTEGER           I,II,III,N(3) , NMAX, J, ERROR, IFAIL, NOUT, INP
      INTEGER           NIN
      PARAMETER         (NMAX=200)
      DOUBLE PRECISION R, SA(2,3), SB(2,3), A(3), H(3), X0(3)
      DOUBLE PRECISION L1, L2, L3, L4
      DOUBLE PRECISION TC(2, NMAX, 3), FC(2, NMAX, 3)
      DOUBLE PRECISION T(2, NMAX, 3), F(2, NMAX, 3),L(NMAX,3)
      DOUBLE PRECISION PHI(NMAX, 3)
      LOGICAL           CROSS, TIBIAL, VARIAT, MECH
      PARAMETER         (L4=100, NOUT=6)
C
C     I,II,III:       COUNTER
C     N(J):           NUMBER OF ITERATIONS
C     NMAX:           MAXIMUM NUMBER OF ITERATIONS
C     J:              COUNTER
C     ERROR:          ERROR INDICATOR
C     IFAIL:          ERROR INDICATOR
C     NOUT:           NUMBER OF OUTPUT DEVICE
C     INP:            INPUT VARIABLE
C     NIN:            NUMBER OF INCREMENTS AT INPUT
C     R:              R = 0: UNCROSSED SYSTEM
C                     R = PI: CROSSED SYSTEM
C     SA(2,J):        X AND Y COORDINATE OF POINT A
C     SB(2,J):        X AND Y COORDINATE OF POINT B
C     A(3):           PARAMETER OF TIBIAL PROFILE
C     H(3):           PARAMETER OF TIBIAL PROFILE
C     X0(3):          PARAMETER OF PARABOLIC TIBIAL PROFILE
C     L1:             LENGTH OF ANTERIOR CRUCIATE LIGAMENT
C     L2:             LENGTH OF POSTERIOR CRUCIATE LIGAMENT
C     L3:             LENGTH OF FEMORAL LINK
C     L4:             LENGTH OF TIBIAL LINK
C     TC(K,L,M):      COORDINATES OF TIBIAL CENTRODE
C                       K: X AND Y COORDINATE
```

```
C                       L: I-TH POINT
C                       M: COUNTER FOR PARAMETER VARIATIONS
C                           M=1: ORIGINAL VALUES
C                           M=2,3: VARIED VALUES
C     FC(K,L,M):      COORDINATES OF FEMORAL CENTRODE
C     T(K,L,M):       COORDINATES OF TIBIAL CONTACT PROFILE
C     F(K,L,M):       COORDINATES OF FEMORAL CONTACT PROFILE
C     L(L,M):         DISTANCE CENTRODE CONTACT PROFILE
C     CROSS:          IF CROSS  IS T(RUE) : CROSSED SYSTEM
C                     IF CROSS  IS F(ALSE): UNCROSSED SYSTEM
C     TIBIAL:         IF TIBIAL IS T(RUE) : TIBIAL PROFILE IS A STRAIGHT
C                                           LINE
C                     IF TIBIAL IS F(ALSE): TIBIAL PROFILE IS A PARABOLE
C     VARIAT:         IF VARIAT IS T(RUE) : PARAMETER VARIATION
C                     IF VARIAT IS F(ALSE): NO PARAMETER VARIATION
C     MECH:           IF MECH   IS F(ALSE : NO MECHANISM CHOSEN YET
C                     IF MECH   IS T(RUE) : MECHANISM CHOSEN YET
C
C     ****************************************
C     ** INITIALISATION OF THE GRAPHICAL MODE **
C     ****************************************
C
      CALL J06VAF(1, NOUT)
      CALL XXXXXX
      OPEN (NOUT, FILE='KNEE.ERR')
      CALL J06WAF
      CALL J06WBF(-25.0D0, 150.0D0, -25.0D0, 150.0D0 ,1)
      CALL J06YLF(5.0D0,0.0D0)
      CALL J06YKF(5.0D0,7.0D0)
C
C
      PRINT 1000
      PRINT 2000,'
      PRINT 2000,'   This program is written for students
      PRINT 2000,'   of the Eindhoven University of Technology.
      PRINT 2000,'   It 's a software program for the simulation
      PRINT 2000,'   of a two-dimensional model of the knee joint.
      PRINT 2000,'   It 's written to show students the effects
      PRINT 2000,'   of parameter variation in a simulation program.
      PRINT 2000,'
      PRINT 1000
      PRINT *,' Press ENTER to continue'
      CALL J06WDF
      PRINT 1000
      PRINT 2000,'
      PRINT 2000,'
      PRINT 2000,'                  WARNING
      PRINT 2000,'
      PRINT 2000,'
      PRINT 2000,'                DO NOT USE STRINGS,
      PRINT 2000,'     use only the numerical part of your keyboard
      PRINT 2000,'
```

```
      PRINT 1000
      PRINT *,' Press ENTER to continue'
      CALL J06WDF
C
C
C     ********************************
C     ** INITIALISATION OF VARIABLE **
C     ********************************
C
      MECH   = .FALSE.
C     ***************
C     ** MAIN MENU **
C     ***************
C
 5    CALL CLEAR
C
      PRINT *,'╔═══════════════════════════════════╗'
      PRINT *,'║ MAIN MENU                         ║'
      PRINT *,'╠═══════════════════════════════════╣'
      PRINT *,'║   1) Input of a mechanism         ║'
      PRINT *,'║   2) Plot results                 ║'
      PRINT *,'║   3) Parameter variation          ║'
      PRINT *,'║   4) Slip ratio                   ║'
      PRINT *,'║   5) Movie                        ║'
      PRINT *,'║   6) Quit                         ║'
      PRINT *,'╚═══════════════════════════════════╝'
      PRINT *,'Enter number of your choice'
C
      READ *, INP
      CALL CLEAR
      IF (INP.LT.1.OR.INP.GT.6) THEN
          CALL CLEAR
          GOTO 5
      ENDIF
      IF (INP.EQ.1) THEN
C
C         ********************************
C         ** CALL FOR SUBROUTINE INPUT **
C         ********************************
C
C         ****************************
C         ** (RE)SET THE VARIABLES **
C         ****************************
C
          MECH = .TRUE.
          VARIAT = .FALSE.
          DO 4 I = 1,3
              DO 3 II = 1,NMAX
                  DO 2 III = 1,2
                      TC(III,II,I) = 0.0D0
                      FC(III,II,I) = 0.0D0
                      T(III,II,I)  = 0.0D0
```

```
                    F(III,II,I)  = 0.0D0
 2                CONTINUE
                  L(II,I)      = 0.0D0
 3            CONTINUE
              N(I) = 0.0D0
 4        CONTINUE
          J = 1
          R = 0.0D0
C
          CALL INPUT(N, NMAX, J, R, SA, SB, A, H, X0, CROSS, TIBIAL)
C
          NIN = N(1)
C
C         *********************************
C         ** CALL FOR SUBROUTINE CENTRO **
C         *********************************
C
          PRINT *,' Calculating centrodes'
          CALL CENTRO(N, NMAX, J, R, SA, SB, L1, L2, L3, L4,
     *                TC, FC, PHI)
C
C
C         *********************************
C         ** CALL FOR SUBROUTINE FEMORA **
C         *********************************
C
          PRINT *,' Calculating femoral contact profile'
          CALL FEMORA(N, NMAX, J, A, H, X0, TC, FC, T,
     *                F, L, TIBIAL, IFAIL, ERROR)
C
          IF (ERROR.EQ.1) THEN
              PRINT 3000
              PRINT 4000, '                                          '
              PRINT 4000, '  WARNING: It is not possible to          '
              PRINT 4000, '  find a femoral contact profile          '
              PRINT 4000, '  which fits.                             '
              PRINT 4000, '  Change the vertical displacement        '
              PRINT 4000, '  H (larger) or the convexity             '
              PRINT 4000, '  A (smaller absolute value).             '
              PRINT 4000, '  (First look at the results.)            '
              PRINT 4000, '                                          '
              PRINT 3000
              PRINT *, ' Press ENTER to continue'
              CALL J06WDF
          ENDIF
          GOTO 5
C
       ELSEIF (INP.EQ.2) THEN
C
C         *********************************
C         ** CALL FOR SUBROUTINE OUTPUT **
C         *********************************
```

```
C
          IF (MECH) THEN
C
              CALL OUTPUT(N, NMAX, J, SA, SB, L4, L, TC, FC, T, F,
     *                    VARIAT)
              GOTO 5
          ELSE
              PRINT 5000
              PRINT 6000, '                                          '
              PRINT 6000, '  No mechanism has been chosen yet        '
              PRINT 6000, '                                          '
              PRINT 5000
              PRINT *, ' Press ENTER to continue'
              CALL J06WDF
              GOTO 5
          ENDIF
C
       ELSEIF (INP.EQ.3) THEN
C
C         ****************************************
C         ** POSSIBILITY FOR PARAMETER VARIATION **
C         ****************************************
C
          VARIAT = .TRUE.
          IF (MECH) THEN
C
C             ******************************
C             ** CALL FOR SUBROUTINE PARA **
C             ******************************
C
              DO 8 I = 2,3
                  DO 7 II = 1,NMAX
                      DO 6 III = 1,2
                          TC(III,II,I) = 0.0D0
                          FC(III,II,I) = 0.0D0
                          T(III,II,I)  = 0.0D0
                          F(III,II,I)  = 0.0D0
 6                    CONTINUE
                      L(II,I)      = 0.0D0
 7                CONTINUE
                  N(I) = NIN
 8            CONTINUE
              J = 1
C
              CALL PARA(N, NMAX, J, ERROR, R, SA, SB, A, H, X0,
     *                  L1, L2, L3, L4, L, TC, FC, T, F, TIBIAL,
     *                  PHI)
              GOTO 5
          ELSE
              PRINT 5000
              PRINT 6000, '                                          '
              PRINT 6000, '  No mechanism has been chosen yet        '
```

```
            PRINT 6000, ' ╚═══════════════════════════════════════════╝'
            PRINT 5000
            PRINT *, ' Press ENTER to continue'
            CALL J06WDF
            GOTO 5
        ENDIF
C
      ELSEIF(INP.EQ.4) THEN
C
C         ******************************
C         ** CALL FOR SUBROUTINE SLIP **
C         ******************************
C
          IF(MECH) THEN
C
            CALL SLIP(N, NMAX, NIN, T, F, VARIAT)
C
            GOTO 5
C
          ELSE
            PRINT 5000
            PRINT 6000, ' ╔═══════════════════════════════════════╗'
            PRINT 6000, ' ║ No mechanism has been chosen yet      ║'
            PRINT 6000, ' ╚═══════════════════════════════════════╝'
            PRINT 5000
            PRINT *, ' Press ENTER to continue'
            CALL J06WDF
            GOTO 5
      ENDIF
C
      ELSEIF (INP.EQ.5) THEN
C
C         ******************************
C         ** CALL FOR SUBROUTINE MOVIE **
C         ******************************
C
          IF (MECH) THEN
C
            CALL  MOVIE(N, NMAX, SA, SB, TC, FC, T, F, VARIAT,
     *                  PHI, R, NIN, L4)
            GOTO 5
          ELSE
            PRINT 5000
            PRINT 6000, ' ╔═══════════════════════════════════════╗'
            PRINT 6000, ' ║ No mechanism has been chosen yet      ║'
            PRINT 6000, ' ╚═══════════════════════════════════════╝'
            PRINT 5000
            PRINT *, ' Press ENTER to continue'
            CALL J06WDF
            GOTO 5
          ENDIF
C
```

```
      ELSE
C
C          **********************
C          ** QUIT THE PROGRAM **
C          **********************
C
           PRINT *,' Press ENTER'
           CALL J06WZF
           STOP ' END OF THE PROGRAM'
      ENDIF
C
 1000 FORMAT (/,/,/,/,/,/,/)
 2000 FORMAT (15X, 51A)
 3000 FORMAT (/,/,/,/,/,/)
 4000 FORMAT (20X, 40A)
 5000 FORMAT (/,/,/,/,/,/,/,/,/,/,/)
 6000 FORMAT (25X, 38A)
C
      END
C
```

Appendix B: Subroutine INPUT

```
      SUBROUTINE INPUT(N, NMAX, J, R, SA, SB, A, H, X0, CROSS, TIBIAL)
C
C     ************************************************************
C     ** THIS SUBROUTINE SUPPLIES THE OTHER ROUTINES WITH THE **
C     **      INPUT DATA AND GIVES THE POSSIBILITY TO QUIT      **
C     ************************************************************
C
C     ******************************************
C     ** DECLARATION OF THE GLOBAL IDENTIFIERS **
C     ******************************************
C
      INTEGER          N(3), NMAX, J
      DOUBLE PRECISION R, SA(2,3), SB(2,3), A(3), H(3), X0(3)
      LOGICAL          CROSS, TIBIAL
C
C     *****************************************
C     ** DECLARATION OF THE LOCAL IDENTIFIERS **
C     *****************************************
C
      INTEGER UU
C
C     UU:     INPUT VARIABLE
C
C     ***********************************************
C     ** CHOISE FOR GIVEN OR SELF MADE MECHANISM **
C     ***********************************************
C
 1    PRINT *,'
      PRINT *,'   TYPE OF MECHANISM                                '
      PRINT *,'
      PRINT *,'   1) Anatomic ( = realistic) dimensions            '
      PRINT *,'   2) Perspex model
      PRINT *,'   3) Self made mechanism                           '
      PRINT *,'
      PRINT *,' Enter number of your choice'
      READ  *, UU
      IF(UU.LT.1.OR.UU.GT.3) THEN
          CALL CLEAR
          GOTO 1
      ENDIF
      IF(UU.EQ.1) THEN
C
C         ***************************************
C         ** PARAMETERS FOR ANATOMIC MECHANISM **
C         ***************************************
C
          CROSS    = .TRUE.
          R        = 3.1415925
          SA(1,J) = 61.0D0
          SA(2,J) = 77.0D0
```

```
          SB(1,J) = 20.0D0
          SB(2,J) = 68.0D0
          TIBIAL  = .FALSE.
          A(J)    = -0.005D0
          X0(J)   = 50.0D0
          H(J)    = 7.0D0
          N(1)    = 200
         GOTO 5
C
      ELSEIF (UU.EQ.2) THEN
C
C         ***********************************
C         ** PARAMETERS FOR PERSPEX MODEL **
C         ***********************************
C
          CROSS   = .TRUE.
          R       = 3.1415925
          SA(1,J) = 91.0D0
          SA(2,J) = 78.0D0
          SB(1,J) = 42.0D0
          SB(2,J) = 78.0D0
          TIBIAL  = .TRUE.
          A(J)    = 0.0D0
          H(J)    = 7.0D0
          N(1)    = 200
          GOTO 5
C
      ENDIF
C
C     *************************
C     ** SELF MADE MECHANISM **
C     *************************
C
C
C     ************************************************
C     ** CHOISE FOR CROSSED OR UNCROSSED MECHANISM **
C     ************************************************
C
 2    CALL CLEAR
      PRINT *,'                                        '
      PRINT *,'   IS THE MECHANISM:                    '
      PRINT *,'                                        '
      PRINT *,'    1) A crossed one                    '
      PRINT *,'    2) An uncrossed one                 '
      PRINT *,'                                        '
      PRINT *,' Enter number of your choice'
      READ  *, UU
      IF(UU.LT.1.OR.UU.GT.2) GOTO 2
      CROSS = .FALSE.
      IF(UU.EQ.1) CROSS = .TRUE.
C
      CALL CLEAR
```

```
C
      IF (CROSS) THEN
          R = 3.14159265
C
C         ***********************************
C         ** PICTURE OF CROSSED MECHANISM **
C         ***********************************
C
          CALL J06YAF(0.0D0, 0.0D0)
          CALL J06YCF(100.0D0, 0.0D0)
          CALL J06YCF(35.0D0, 70.0D0)
          CALL J06YCF(75.0D0, 85.0D0)
          CALL J06YCF(0.0D0, 0.0D0)
          CALL J06YAF(80.0D0, 90.0D0)
          CALL J06YHF('A',1)
          CALL J06YAF(25.0D0, 75.0D0)
          CALL J06YHF('B',1)
          CALL J06YAF(10.0D0,-30.0D0)
          CALL J06YHF('CROSSED MECHANISM',17)
C
      ELSE
           R = 0
C
C          *************************************
C          ** PICTURE OF UNCROSSED MECHANISM **
C          *************************************
C
          CALL J06YAF(0.0D0, 0.0D0)
          CALL J06YCF(100.0D0, 0.0D0)
          CALL J06YCF(75.0D0, 85.0D0)
          CALL J06YCF(35.0D0, 70.0D0)
          CALL J06YCF(0.0D0, 0.0D0)
          CALL J06YAF(80.0D0, 90.0D0)
          CALL J06YHF('B',1)
          CALL J06YAF(25.0D0, 75.0D0)
          CALL J06YHF('A',1)
          CALL J06YAF(5.0D0,-30.0D0)
          CALL J06YHF('UNCROSSED MECHANISM',19)
C
C
      ENDIF
C
      CALL J06YAF(-10.0D0, -10.0D0)
      CALL J06YHF('O=(0, 0)',8)
      CALL J06YAF(100.0D0, -10.0D0)
      CALL J06YHF('P=(100, 0)',10)
      CALL J06YAF(20.0D0, 0.0D0)
      CALL J06YCF(17.0D0, 2.0D0)
      CALL J06YCF(17.0D0, -2.0D0)
      CALL J06YCF(20.0D0, 0.0D0)
      CALL J06YAF(20.0D0, 2.0D0)
      CALL J06YHF('X',1)
```

```
      CALL J06YAF(0.0D0, 0.0D0)
      CALL J06YCF(0.0D0, 20.0D0)
      CALL J06YCF(2.0D0, 17.0D0)
      CALL J06YCF(-2.0D0, 17.0D0)
      CALL J06YCF(0.0D0, 20.0D0)
      CALL J06YAF(-10.0D0, 20.0D0)
      CALL J06YHF('Y',1)
C
C     ************************************
C     ** INPUT OF COORDINATES OF A AND B **
C     ************************************
C
      PRINT *,' Give the coordinates of hinges A and B'
      PRINT *,' X-coordinate of A:'
      READ  *, SA(1,J)
      PRINT *,' Y-coordinate of A:'
      READ  *, SA(2,J)
      PRINT *,' X-coordinate of B:'
      READ  *, SB(1,J)
      PRINT *,' Y-coordinate of B:'
      READ  *, SB(2,J)
C
C
C     ************************************************
C     ** CHOISE AND INPUT OF TIBIAL CONTACT PROFILE **
C     ************************************************
C
 3    CALL CLEAR
      PRINT *,'
      PRINT *,'    THE TIBIAL PROFILE                  '
      PRINT *,'
      PRINT *,'    1) is a straight line               '
      PRINT *,'    2) is a parabole                    '
      PRINT *,'
      PRINT *,' Enter number of your choice'
      READ  *, UU
      IF(UU.LT.1.OR.UU.GT.2) GOTO 3
      TIBIAL = .FALSE.
      IF(UU.EQ.1) TIBIAL = .TRUE.
C
      CALL CLEAR
      IF (TIBIAL) THEN
          PRINT *,'
          PRINT *,'    A straight line is described by    '
          PRINT *,'
          PRINT *,'          " Y = A*X + H "              '
          PRINT *,'
          PRINT *,' GIVE A (-0.5 ≤ A ≤ 0.5):'
          READ  *, A(J)
          PRINT *,' GIVE H (-50 ≤ H ≤ 50):'
          READ  *, H(J)
      ELSE
```

```
          PRINT *,'                                          '
          PRINT *,'   A parabole is described by             '
          PRINT *,'                                          '
          PRINT *,'     " Y = A(X-X0)^2 + H "                '
          PRINT *,'                                          '
          PRINT *,' GIVE A (-0.01 ≤ A ≤ 0.01):'
          READ  *, A(J)
          PRINT *,' GIVE X0 (0 ≤ X0 ≤ 100):'
          READ  *, X0(J)
          PRINT *,' GIVE H (-50 ≤ H ≤ 50):'
          READ  *, H(J)
        ENDIF
        CALL CLEAR
C
C       ************************************************
C       ** INPUT OF NUMBER OF INCREMENTS (= ACCURACY) **
C       ************************************************
C
 4      PRINT *,' Give the number of increments (10 - 200)'
        READ  *, N(J)
        IF(N(J).GT.NMAX) GOTO 4
C
C
 5      END
C
```

Appendix C: Subroutine CENTRO

```
      SUBROUTINE CENTRO(N, NMAX, J, R, SA, SB,
     *                  L1, L2, L3, L4, TC, FC, PHI)
C
C     ****************************************************************
C     ** THIS SUBROUTINE CALCULATES THE TIBIAL AND FEMORAL CENTRODE. **
C     ** THE X AND Y COORDINATES ARE STORED IN THE ARRAYS TC AND FC. **
C     ****************************************************************
C
C     ****************************************
C     ** DECLARATION OF THE GLOBAL PARAMETERS **
C     ****************************************
C
      INTEGER           N(3), NMAX, J
      DOUBLE PRECISION SA(2,3), SB(2,3), R
      DOUBLE PRECISION L1, L2, L3, L4, TC(2,NMAX,3), FC(2,NMAX,3)
      DOUBLE PRECISION PHI(NMAX,3)
C
C     ****************************************
C     ** DECLARATION OF THE LOCAL PARAMETERS **
C     ****************************************
C
      INTEGER           I, NNMAX
      PARAMETER         (NNMAX = 200)
      DOUBLE PRECISION ALPHA(NNMAX), THETA(NNMAX), SALPHA
      DOUBLE PRECISION CAPHI, CBPHI, CCPHI, TEK, CATHET, GAMMA
      DOUBLE PRECISION CBTHET, CCTHET, C1, BETA, L1AA, DELTA, PI
      PARAMETER         (PI=3.14159265)
C
C     I:              COUNTER
C     NNMAX:          MAXIMAL NUMBER OF INCREMENTS
C     ALPHA(NNMAX):   ANGLE BETWEEN TIBIAL LINK AND FEMORAL LINK
C     SALPHA          ANGLE ALPHA AT THE GIVEN POSITION (RADIANS)
C     PHI(NNMAX):     ANGLE BETWEEN TIBIAL LINK AND O-A
C     THETA(NNAMX):   ANGLE BETWEEN TIBIAL LINK AND P-B
C     CAPHI:          VARIABLE
C     CBPHI:          VARIABLE
C     CCPHI:          VARIABLE
C     TEK:            VARIABLE
C     CATHET:         VARIABLE
C     GAMMMA:         VARIABLE
C     CBTHET:         VARIABLE
C     CCTHET:         VARIABLE
C     C1:             DISTANCE POINT O TO TIBIAL CENTRODE
C     BETA:           ANGLE ALPHA - SALPHA
C     L1AA:           DISTANCE FROM A TO FC
C     DELTA:          ANGLE PHI - BETA
C     PI:             3.14159265
C
C
```

```
C      ************************************
C      ** CALCULATION OF THE ANGLE SALPHA **
C      ************************************
C
       SALPHA = R + ATAN ((SB(2,J)-SA(2,J))/(SB(1,J)-SA(1,J)))
C
C      ********************************************
C      ** CALCULATION OF THE LENGTHS OF THE LINKS **
C      ********************************************
C
       L1 = SQRT(SA(1,J)*SA(1,J) + SA(2,J)*SA(2,J))
       L2 = SQRT((L4-SB(1,J))*( L4-SB(1,J))+SB(2,J)*SB(2,J))
       L3 = SQRT((SA(1,J)-SB(1,J))*(SA(1,J)-SB(1,J))+(SA(2,J)-SB(2,J))*
      *           (SA(2,J)-SB(2,J)))
C
C
       I = 0
 1     I = I + 1
C
C      ************************************************
C      ** CALCULATION OF THE ANGLE ALPHA (IN RADIANS) **
C      ************************************************
C
       ALPHA(I) = ((50./180)*PI)+R-(((I-1.)/(N(J)-1))*(140./180)*PI)
C
C      **********************************
C      ** CALCULATION OF THE ANGLE PHI **
C      **********************************
C
       CAPHI = SIN(ALPHA(I))
       CBPHI = COS(ALPHA(I)) - L4/L3
       CCPHI = (L4/L1)*COS(ALPHA(I)) +
      *        (-L1*L1 + L2*L2 - L3*L3 - L4*L4)/(2*L1*L3)
C
       PHI(I,J)= 2*ATAN((CAPHI - SQRT(CAPHI*CAPHI +
      *                               CBPHI*CBPHI -
      *                               CCPHI*CCPHI)) /
      *                              (CBPHI + CCPHI))
       IF(I.GT.2) THEN
           TEK = (PHI(I,J)-PHI(I-1,J))*(PHI(I-1,J)-PHI(I-2,J))
           IF(TEK.LT.0) THEN
               N(J)  = I-1
               CALL CLEAR
               GAMMA = -((ALPHA(I-1)-R)*180.0D0/PI-50)
       PRINT 2000
       PRINT 1000,'╔═══════════════════════════════════════════════════╗'
       PRINT 1000,'║                                                   ║'
       PRINT 1000,'║  It is not possible to calculate the full range   ║'
       PRINT 1000,'║  of the flexion angle (gamma) for this system.    ║'
       PRINT 1000,'║  The program will calculate the centrodes and     ║'
       PRINT 1000,'║  the femoral contact profile for:                 ║'
       PRINT 1000,'║                                                   ║'
       PRINT 3000, GAMMA
```

```
      PRINT 1000,'╟                                                    ╢'
      IF(J.EQ.1) THEN
      PRINT 1000,'╟  The mechanism will be drawn in WHITE              ╢'
      ELSEIF(J.EQ.2) THEN
      PRINT 1000,'╟  The mechanism will be drawn in RED                ╢'
      ELSE
      PRINT 1000,'╟  The mechanism will be drawn in GREEN              ╢'
      ENDIF
      PRINT 1000,'╟                                                    ╢'
      PRINT 1000,'╚════════════════════════════════════════════════════╝'
      PRINT 2000
      PRINT *,'Press ENTER to continue'
              CALL J06WDF
              GOTO 2
          ENDIF
      ENDIF
C
C     ************************************
C     ** CALCULATION OF THE ANGLE THETA **
C     ************************************
C
      CATHET =  SIN(ALPHA(I))
      CBTHET =  COS(ALPHA(I)) - L4/L3
      CCTHET =  (-L4/L2)*COS(ALPHA(I)) -
     *          (L1*L1 - L2*L2 - L3*L3 -L4*L4)/(2*L2*L3)
C
      THETA(I)= 2*ATAN((CATHET - SQRT(CATHET*CATHET +
     *                                CBTHET*CBTHET -
     *                                CCTHET*CCTHET)) /
     *                               (CBTHET + CCTHET))
      IF(I.GT.2) THEN
          TEK = (THETA(I)-THETA(I-1))*(THETA(I-1)-THETA(I-2))
          IF(TEK.LT.0) THEN
              N(J) = I-1
C
               CALL CLEAR
               GAMMA = -((ALPHA(I-1)-R)*180.0D0/PI-50)
      PRINT 2000
      PRINT 1000,'╔════════════════════════════════════════════════════╗'
      PRINT 1000,'╟  It is not possible to calculate the full range    ╢'
      PRINT 1000,'╟  of the flexion angle (gamma) for this system.     ╢'
      PRINT 1000,'╟  The program will calculate the centrodes and      ╢'
      PRINT 1000,'╟  the femoral contact profile for:                  ╢'
      PRINT 1000,'╟                                                    ╢'
      PRINT 3000, GAMMA
      PRINT 1000,'╟                                                    ╢'
      IF(J.EQ.1) THEN
      PRINT 1000,'╟  This mechanism will be drawn in WHITE             ╢'
      ELSEIF(J.EQ.2) THEN
      PRINT 1000,'╟  This mechanism will be drawn in RED               ╢'
      ELSE
      PRINT 1000,'╟  This mechanism will be drawn in GREEN             ╢'
```

```
      ENDIF
      PRINT 1000,'╔══════════════════════════════════════════════════╗'
      PRINT 1000,'╚══════════════════════════════════════════════════╝'
      PRINT 2000
      PRINT *,'Press ENTER to continue'
              CALL J06WDF
              GOTO 2
          ENDIF
      ENDIF
C
C     ************************************
C     ** CALCULATION OF TIBIAL CENTRODE **
C     ************************************
C
      C1 = L4*SIN(THETA(I))/SIN(THETA(I) - PHI(I,J))
C
      TC(1,I,J) = C1*COS(PHI(I,J))
      TC(2,I,J) = C1*SIN(PHI(I,J))
C
C
C     ************************************
C     ** CALCULATION OF FEMORAL CENTRODE **
C     ************************************
C
      BETA = ALPHA(I) - SALPHA
      DELTA = PHI(I,J) - BETA
      L1AA = ABS(L1 - C1)
C
      FC(1,I,J) = SA(1,J) + L1AA*COS(DELTA + R)
      FC(2,I,J) = SA(2,J) + L1AA*SIN(DELTA + R)
      IF(I.LT.N(J)) GO TO 1
C
 1000 FORMAT (15X, 52A)
 2000 FORMAT (/,/,/,/)
 3000 FORMAT (15X , '‖', 15X, '0° ≤ gamma ≤', 1X, F6.1, '°', 14X, '‖')
C
 2    END
C
```

Appendix D: Subroutine FEMORA

```
      SUBROUTINE FEMORA(N, NMAX, J, A, H, X0, TC, FC,
     *                  T, F, L, TIBIAL, IFAIL, ERROR)
C
C     ********************************************************
C     ** THIS SUBROUTINE CALCULATES THE POINTS OF THE TIBIAL **
C     ** AND FEMORAL CONTACT PROFILE. THE COORDINATES ARE    **
C     ** STORED IN THE VARIABLES T AND F. THE RELEVANT       **
C     ** DISTANCE FROM A POINT ON THE CENTRODE TO A POINT    **
C     ** ON THE CONTACT PROFILE IS STORED IN L.              **
C     ********************************************************
C
C
C     ******************************************
C     ** DECLARATION OF THE GLOBAL IDENTIFIERS **
C     ******************************************
C
      INTEGER           N(3), NMAX, J, IFAIL, ERROR
      DOUBLE PRECISION A(3), H(3), X0(3), TC(2,NMAX,3), FC(2,NMAX,3)
      DOUBLE PRECISION T(2,NMAX,3), F(2,NMAX,3), L(NMAX,3)
      LOGICAL           TIBIAL
C
C
C     ******************************************
C     ** DECLARATION OF THE LOCAL IDENTIFIERS **
C     ******************************************
C
      INTEGER           I, SP, U, WW
      DOUBLE PRECISION NT(2), CT(2), CF(2), CHI, S
      DOUBLE PRECISION E, DD, PI
      PARAMETER         (PI = 3.141592653)
C
C     I:         COUNTER
C     SP:        STARTING VALUE OF I
C     U:         U = 1 OR -1
C     WW:        WW = 1 OR -1
C     NT(2):     THE COMPONENTS OF THE VECTOR Nt
C     CT(2):     THE COMPONENTS OF THE VECTOR Ct
C     CF(2):     THE COMPONENTS OF THE VECTOR Cf
C     CHI:       THE ANGLE BETWEEN Nt AND Ct
C     S:         THE LENGTHS OF THE VECTOR Cf
C     E:         INTER VARIABLE
C     DD:        INTER VARIABLE
C     PI:        3.141592653

      ERROR = 0
C
C     ****************************************************
C     **     CALCULATION OF FEMORAL CONTACT PROFILE     **
C     ** WITH TIBIAL CONTACT PROFILE IS A STRAIGHT LINE **
C     ****************************************************
```

```
C
      IF (TIBIAL) THEN
          DO 1 I=2, N(J)-1
              T(1,I,J)    = (A(J)*TC(2,I,J) - A(J)*H(J) + TC(1,I,J)) /
     *                      (1 + A(J)*A(J))
              T(2,I,J)    = A(J)*T(1,I,J) + H(J)
              L(I,J)      = (TC(2,I,J) - A(J)*TC(1,I,J) - H(J))/
     *                      (SQRT(1 + A(J)*A(J)))
              NT(1)       = -A(J) / SQRT(1 + A(J)*A(J))
              NT(2)       = 1 / SQRT(1 + A(J)*A(J))
              WW = 1
              IF (TC(1,I-1,J).GT.TC(1,I+1,J)) WW = -1
              CT(1)       = WW*(TC(1,I+1,J)-TC(1,I-1,J))
              CT(2)       = WW*(TC(2,I+1,J)-TC(2,I-1,J))
              CF(1)       = WW*(FC(1,I+1,J)-FC(1,I-1,J))
              CF(2)       = WW*(FC(2,I+1,J)-FC(2,I-1,J))
              CHI         = PI-ACOS(((NT(1)*CT(1))+(NT(2)*CT(2) ) ) /
     *                      (SQRT(CT(1)*CT(1) + CT(2)*CT(2)) ) )
              S           = SQRT(CF(1)*CF(1) + CF(2)*CF(2))
              F(1,I,J)    = FC(1,I,J)+L(I,J)/S*
     *                      (CF(1)*COS(CHI) + CF(2)*SIN(CHI))
              F(2,I,J)    = FC(2,I,J)+L(I,J)/S*
     *                      (-CF(1)*SIN(CHI) + CF(2)*COS(CHI))
 1        CONTINUE
C
      ELSE
C
C         ***********************************************
C         **  CALCULATION OF FEMORAL CONTACT PROFILE   **
C         ** WITH TIBIAL CONTACT PROFILE IS A PARABOLE **
C         ***********************************************
C
C         ***************************************************
C         ** DETERMINATION OF STARTING POINT FOR ITERATION **
C         ***************************************************
C
          I = 2
          DD = ABS(X0(J) - TC(1,I,J))
 2        I = I+1
          IF (I.GT.N(J)-1) THEN
              SP = N(J)-1
              GOTO 3
          ENDIF
          E = DD
          DD = ABS(X0(J) - TC(1,I,J))
          IF (DD.LT.E) THEN
              GOTO 2
          ELSE
              SP = I-1
          ENDIF
C
```

```
C          ********************************************
C          ** CALCULATION OF FEMORAL CONTACT PROFILE **
C          **    FOR I = SP TO 2 (BACKWARD)          **
C          ********************************************
C
 3         T(1,SP+1,J) = X0(J)
           U = 1
           I = SP
 4         IF (I.GE.2) THEN
               CALL CALC(NMAX, J, I, U, A, H, X0, TC, FC,
     *                   T, F, L, IFAIL, ERROR)
               IF (IFAIL.NE.0) THEN
                   PRINT *,' ERROR SUBROUTINE C02AEF, IFAIL= ',IFAIL
                   GOTO 6
               ENDIF
               I = I - 1
               GOTO 4
           ENDIF
C
C          ********************************************
C          ** CALCULATION OF FEMORAL CONTACT PROFILE **
C          **    FOR I = SP TO N-1 (FORWARD)         **
C          ********************************************
C
           I = SP + 1
           U = -1
 5         IF (I.LE.N(J)-1) THEN
               CALL CALC(NMAX, J, I, U, A, H, X0, TC, FC,
     *                   T, F, L, IFAIL, ERROR)
               IF (IFAIL.NE.0) THEN
                   PRINT *,' ERROR SUBROUTINE C02AEF, IFAIL= ',IFAIL
                   GOTO 6
               ENDIF
               I = I + 1
               GOTO 5
           ENDIF
       ENDIF
 6     END
C
       SUBROUTINE CALC(NMAX, J, I, U, A, H, X0, TC, FC,
      *                T, F, L, IFAIL, ERROR)
C
C      ************************************************************
C      ** SUBROUTINE FOR CALCULATION OF POINT OF FEMORAL CENTRODE **
C      ************************************************************
C
C      ******************************************
C      ** DECLARATIONS OF THE GLOBAL IDENTIFIERS **
C      ******************************************
C
       INTEGER           NMAX, J, I, U, IFAIL, ERROR
       DOUBLE PRECISION A(3), H(3), X0(3), TC(2,NMAX,3), FC(2,NMAX,3)
```

```
      DOUBLE PRECISION T(2,NMAX,3), F(2,NMAX,3), L(NMAX,3)
C
C
C     ****************************************
C     ** DECLARATION OF THE LOCAL IDENTIFIERS **
C     ****************************************
C
      INTEGER           Q, K, D, TT, WW
      DOUBLE PRECISION AA(4), Z(2,3), XX(3), NT(2), CT(2), CF(2)
      DOUBLE PRECISION CHI, S, ZERO, WOR, DIF(3), W, ZZ(3), ZZZ(3), PI
      PARAMETER         (ZERO = 0.0D0, Q = 4, W=1.0 D -8)
      PARAMETER         (PI = 3.141592653)
C
C     Q:          NUMBER OF COEFFICIENTS (Q=4)
C     K:          NUMBER OF REAL ROOTS OF THIRD ORDER POLYNOME
C     D:          COUNTER
C     TT:         COUNTER
C     WW:         WW = 1 OR -1
C     AA(4):      PARAMETERS OF THIRD ORDER POLYNOME
C     Z(2,3):     ROOTS OF THIRD ORDER POLYNOME
C     XX(3):      INTERIM VARIABLE (REAL ROOTS OF THIRD ORDER POLYNOME)
C     NT(2):      THE COMPONENTS OF THE VECTOR Nt
C     CT(2):      THE COMPONENTS OF THE VECTOR Ct
C     CF(2):      THE COMPONENTS OF THE VECTOR Cf
C     CHI:        THE ANGLE BETWEEN Nt AND Ct
C     S:          THE LENGTHS OF THE VECTOR Cf
C     ZERO:       = 0.0 D 0
C     WOR:        INTERIM VARIABLE
C     DIF(3):     INTERIM VARIABLE
C     W:          TOLERANCE FOR SUBROUTINE C02AEF
C     ZZ(3):      REAL PART OF SOLUTION THIRD ORDER POLYNOME
C     ZZZ(3):     IMAGINARY PART OF SOLUTION THIRD ORDER POLYNOME
C     PI:         3.141592653
C
C     ********************************************
C     ** CALCULATIONS FOR PARAMETERS OF POLYNOME **
C     ********************************************
C
      AA(1) = 2*A(J)*A(J)
      AA(2) = -6*A(J)*A(J)*X0(J)
      AA(3) = 1-2*A(J)*TC(2,I,J)+2*A(J)*H(J)+6*A(J)*A(J)*X0(J)*X0(J)
      AA(4) = 2*A(J)*X0(J)*TC(2,I,J)-2*A(J)*H(J)*X0(J)-
     *        2*A(J)*A(J)*X0(J)*X0(J)*X0(J) - TC(1,I,J)
C
C
C     ************************************************
C     ** CALCULATION OF ROOTS OF THIRD ORDER POLYNOME **
C     ************************************************
C
      ZZ(1) = 1.0D0
      ZZZ(1)= 1.0D0
      IFAIL = 0
```

```
      CALL C02AEF (AA,Q,ZZ,ZZZ,W,IFAIL)
      DO 1 TT=1,3
          Z(1,TT)=ZZ(TT)
          Z(2,TT)=ZZZ(TT)
 1    CONTINUE
      IF (IFAIL.NE.0) THEN
          PRINT *,' ERROR SUBROUTINE C02AEF, IFAIL= ',IFAIL
          GOTO 3
      ENDIF
C
C     ****************************
C     ** SELECTION OF REAL ROOTS **
C     ****************************
C
      K = 0
      DO 2 D=1,3
          IF (Z(2,D).EQ.ZERO) THEN
              K = K + 1
              XX(K) = Z(1,D)
          ENDIF
 2    CONTINUE
      IF(K.GE.2) ERROR = 1
C
C     ************************************
C     ** SELECTION OF RELEVANT SOLUTION **
C     ************************************
C
      IF (K.EQ.1) THEN
          T(1,I,J)= XX(1)
      ELSEIF (K.EQ.2) THEN
          DIF(1) = ABS(XX(1) - T(1,I+U,J))
          DIF(2) = ABS(XX(2) - T(1,I+U,J))
          IF (DIF(1).LE.DIF(2)) THEN
              T(1,I,J) = XX(1)
          ELSE
              T(1,I,J) = XX(2)
          ENDIF
      ELSE
          DIF(1) = ABS(XX(1) - T(1,I+U,J))
          DIF(2) = ABS(XX(2) - T(1,I+U,J))
          DIF(3) = ABS(XX(3) - T(1,I+U,J))
          IF (DIF(1).LE.DIF(2).AND.DIF(1).LE.DIF(3)) THEN
              T(1,I,J) = XX(1)
          ELSEIF (DIF(2).LE.DIF(3).AND.DIF(2).LE.DIF(1)) THEN
              T(1,I,J) = XX(2)
          ELSE
              T(1,I,J) = XX(3)
          ENDIF
      ENDIF
C
C
```

```
C     ******************************************************
C     ** CALCULATION OF POINT OF FEMORAL CONTACT PROFILE **
C     ******************************************************
C
C
      WOR        = SQRT(4*A(J)*A(J)*(T(1,I,J)-X0(J))*(T(1,I,J)-X0(J))+1)
      T(2,I,J)   = A(J)*(T(1,I,J)-X0(J))*(T(1,I,J)-X0(J))+H(J)
      L(I,J)     = (TC(2,I,J) - A(J)*(T(1,I,J)-X0(J))*(T(1,I,J)-
     *             X0(J))-H(J))*WOR
      NT(1)      = -2*A(J)*(T(1,I,J)-X0(J))/WOR
      NT(2)      = 1/WOR
      WW         = 1
      IF (TC(1,I-1,J).GT.TC(1,I+1,J)) WW = -1
      CT(1)      = WW*(TC(1,I+1,J)-TC(1,I-1,J))
      CT(2)      = WW*(TC(2,I+1,J)-TC(2,I-1,J))
      CF(1)      = WW*(FC(1,I+1,J)-FC(1,I-1,J))
      CF(2)      = WW*(FC(2,I+1,J)-FC(2,I-1,J))
      CHI        = PI - ACOS( ( (NT(1)*CT(1)) + (NT(2)*CT(2)) )/
     *             ( SQRT(CT(1)*CT(1) + CT(2)*CT(2)) ) )
      S          = SQRT(CF(1)*CF(1) + CF(2)*CF(2))
      F(1,I,J)   = FC(1,I,J)+L(I,J)/S*
     *             (CF(1)*COS(CHI) + CF(2)*SIN(CHI))
      F(2,I,J)   = FC(2,I,J)+L(I,J)/S*
     *             (-CF(1)*SIN(CHI) + CF(2)*COS(CHI))
 3    END
C
```

Appendix E: Subroutine OUTPUT

```
      SUBROUTINE OUTPUT(N, NMAX, J, SA, SB, L4, L, TC, FC, T, F, VARIAT)
C
C     ****************************************************************
C     ** THIS SUBROUTINE VISUALISES THE RESULTS OF THE CALCULATIONS **
C     ****************************************************************
C
C     ****************************************
C     ** DECLARATION OF THE GLOBAL IDENTIFIERS **
C     ****************************************
C
      INTEGER           N(3), NMAX, J
      DOUBLE PRECISION SA(2,3), SB(2,3), L4, L(NMAX, 3), TC(2, NMAX, 3)
      DOUBLE PRECISION FC(2, NMAX, 3), T(2, NMAX, 3), F(2, NMAX, 3)
      LOGICAL           VARIAT
C
C     ****************************************
C     ** DECLARATION OF THE LOCAL IDENTIFIERS **
C     ****************************************
C
      INTEGER           I, DAT, II
      DOUBLE PRECISION ZERO
      PARAMETER         (ZERO = 0.0D0, DAT = 7)
C
C     I:          COUNTER
C     II:         COUNTER
C     DAT:        NUMBER OF OUTPUT DEVICE
C     ZERO:       = 0.0D0

C     ********************************
C     ** SAVING DATA TO FILE KNEE.DAT **
C     ********************************
C
C      OPEN (DAT, FILE='KNEE.DAT', STATUS='UNKNOWN')
C     DO 2 II = 1,J
C          WRITE (DAT, 1000) SA(1,J), SA(2,J), SB(1,J), SB(2,J)
C          WRITE (DAT, 3000)
C          DO 1 I = 2, N(J)-1
C              WRITE (DAT, 2000) I, TC(1,I,J), FC(1,I,J), T(1,I,J),
C     *        F(1,I,J), L(I, J), TC(2,I,J), FC(2,I,J), T(2,I,J),
C     *        F(2,I,J)
C 1        CONTINUE
C 2    CONTINUE
C
C
C     ********************************************
C     ** SETTING OF CURRENT LINESTYLE AND COLOUR **
C     ********************************************
C
C     J06YQF: SETS THE CURRENT COLOUR
C     J06YRF: SETS THE CURRENT LINESTYLE
```

```
C
      J = 1
C
 3    CALL J06YQF(J)
      CALL J06YRF(J)
C
C
C     ********************************
C     ** DRAWING OF TIBIAL CENTRODE **
C     ********************************
C
C     J06YAF:   MOVES PEN TO A GIVEN POINT
C     J06YCF:   DRAWS LINE BETWEEN TWO POINTS
C
      CALL J06YAF(TC(1,1,J),TC(2,1,J))
      DO 4 I=2, N(J)
          CALL J06YCF(TC(1,I,J),TC(2,I,J))
 4    CONTINUE
C
C     ********************************
C     ** DRAWING OF FEMORAL CENTRODE **
C     ********************************
C
      IF(.NOT.VARIAT) CALL J06YQF(2)
C
      CALL J06YAF(FC(1,1,J),FC(2,1,J))
      DO 5 I=2,N(J)
          CALL J06YCF(FC(1,I,J),FC(2,I,J))
 5    CONTINUE
C
C     ***************************************
C     ** DRAWING OF TIBIAL CONTACT PROFILE **
C     ***************************************
C
      IF(.NOT.VARIAT) CALL J06YQF(3)
C
      CALL J06YAF(T(1,2,J),T(2,2,J))
      DO 6 I=3,N(J)-1
          CALL J06YCF(T(1,I,J),T(2,I,J))
 6    CONTINUE
C
C     *****************************************
C     ** DRAWING OF FEMORAL CONTACT PROFILE **
C     *****************************************
C
      IF(.NOT.VARIAT) CALL J06YQF(4)
C
      CALL J06YAF(F(1,2,J),F(2,2,J))
      DO 7 I=3,N(J)-1
          CALL J06YCF(F(1,I,J),F(2,I,J))
 7    CONTINUE
C
```

```
C     ****************************
C     ** DRAWING INPUT SITUATION **
C     ****************************
C
      IF(.NOT.VARIAT) CALL J06YQF(5)
C
      CALL J06YAF(ZERO, ZERO)
      CALL J06YCF(L4, ZERO)
      CALL J06YCF(SB(1,J), SB(2,J))
      CALL J06YCF(SA(1,J), SA(2,J))
      CALL J06YCF(ZERO, ZERO)
C
C
      IF (VARIAT) THEN
          IF (J.LT.3) THEN
              J=J+1
              GOTO 3
          ENDIF
      ENDIF
C
      IF(.NOT.VARIAT) THEN
          PRINT *,'YELLOW = Input situation of the mechanism'
          PRINT *,'WHITE  = Tibial  centrode'
          PRINT *,'RED    = Femoral centrode'
          PRINT *,'GREEN  = Tibial  contact profile'
          PRINT *,'BLUE   = Femoral contact profile'
          PRINT 4000
      ELSE
          PRINT 5000
      ENDIF
C
      PRINT *,' Press ENTER to continue'
      CALL J06WDF
C
 1000 FORMAT(F9.4, 5X, F9.4, 5X, F9.4, 5X, F9.4)
 2000 FORMAT(I3,4X, 'X-COORD.', 4X, F8.3, 4X, F8.3,
     *       4X, F8.3, 4X, F8.3, 4X, F8.3, /, 7X, 'Y-COORD.',
     *       4X, F8.3, 4X, F8.3, 4X, F8.3, 4X, F8.3)
 3000 FORMAT(2X,'I',21X,'TC',10X,'FC',10X,'T',10X,'F',10X,'L',/)
 4000 FORMAT(/,/,/,/,/,/,/,/,/,/,/,/,/,/,/,/,/,/,/,/)
 5000 FORMAT(/,/,/,/,/,/,/,/,/,/,/,/,/,/,/,/,/,/,/,/,/,/,/,/,/,/,/,/)
C
      END
C
```

Appendix F: Subroutine PARA

```
      SUBROUTINE PARA(N, NMAX, J, ERROR, R, SA, SB, A, H, XO,
     *                L1, L2, L3, L4, L, TC, FC, T, F, TIBIAL,
     *                PHI)
C
C     ************************************************************
C     ** THIS SUBROUTINE CALCULATES THE EFFECTS OF VARIATIONS   **
C     **   OF SEVERAL PARAMETERS. THIS IS DONE BY MAKING THEM   **
C     **      A CERTAIN PERCENTAGE C SMALLER OR GREATER         **
C     ************************************************************
C
C     ******************************************
C     ** DECLARATION OF THE GLOBAL IDENTIFIERS **
C     ******************************************
C
      INTEGER          N(3), NMAX, J, ERROR
      DOUBLE PRECISION R, SA(2,3), SB(2,3), A(3), H(3), XO(3)
      DOUBLE PRECISION L1, L2, L3, L4, L(NMAX,3)
      DOUBLE PRECISION TC(2, NMAX, 3), FC(2, NMAX, 3)
      DOUBLE PRECISION T(2, NMAX, 3), F(2, NMAX, 3)
      DOUBLE PRECISION PHI(NMAX,3)
      LOGICAL          TIBIAL
C
C     ******************************************
C     ** DECLARATION OF THE LOCAL IDENTIFIERS **
C     ******************************************
C
      INTEGER          I, CHAR, IFAIL, UU
      DOUBLE PRECISION C(2), CC(2), CCC(2)
C
C     I:               COUNTER
C     CHAR:            INPUT VARIABLE
C     IFAIL:           ERROR INDICATOR
C     UU:              INPUT VARIABLE
C     C(2):            VALUE FOR PARAMETER VARIATION
C     CC(2):           VALUE FOR PARAMETER VARIATION
C     CCC(2):          VALUE FOR PARAMETER VARIATION
C
      C(1)   = -0.1D0
      C(2)   = 0.1D0
      CC(1)  = -10.0D0
      CC(2)  = 10.0D0
      CCC(1)= -0.0025
      CCC(2)= 0.0025
C
C     ****************************************
C     ** CHOISE OF PARAMETER TO BE VARIATED **
C     ****************************************
C
 1    PRINT *,' ╔═══════════════════════════════════╗'
      PRINT *,' ║   PARAMETER VARIATION             ║'
```

```
      PRINT *,'╔═══════════════════════════════════╗'
      PRINT *,'║  1) Elements are rigid            ║'
      PRINT *,'║  2) Elements are flexible         ║'
      PRINT *,'╚═══════════════════════════════════╝'
      PRINT *,' Enter number of your choice'
      READ  *, UU
      CALL CLEAR
      IF(UU.LT.1.OR.UU.GT.2) THEN
          GOTO 1
      ENDIF
C
      IF(UU.EQ.2) THEN
C
         PRINT 1000
         PRINT 2000,'╔═══════════════════════════════════════════════════╗'
         PRINT 2000,'║ This part has not yet been completed.             ║'
         PRINT 2000,'║ Therefore you will have to use rigid elements     ║'
         PRINT 2000,'╚═══════════════════════════════════════════════════╝'
         PRINT 1000
         PRINT *,'Press ENTER to continue'
         CALL J06WDF
         GOTO 1
      ENDIF
      CALL J06WBF(-25.0D0, 250.0D0, -25.0D0, 250.0D0 ,1)
C
      IF (R.GT.1) THEN
C
C          **********************************
C          ** PICTURE OF CROSSED MECHANISM **
C          **********************************
C
           CALL J06YAF(0.0D0, 0.0D0)
           CALL J06YCF(100.0D0, 0.0D0)
           CALL J06YCF(35.0D0, 70.0D0)
           CALL J06YCF(75.0D0, 85.0D0)
           CALL J06YCF(0.0D0, 0.0D0)
           CALL J06YAF(80.0D0, 90.0D0)
           CALL J06YHF('A',1)
           CALL J06YAF(25.0D0, 75.0D0)
           CALL J06YHF('B',1)
           CALL J06YAF(10.0D0,-30.0D0)
           CALL J06YHF('CROSSED MECHANISM',17)
C
      ELSE
C
C          ************************************
C          ** PICTURE OF UNCROSSED MECHANISM **
C          ************************************
C
           CALL J06YAF(0.0D0, 0.0D0)
           CALL J06YCF(100.0D0, 0.0D0)
           CALL J06YCF(75.0D0, 85.0D0)
```

```
          CALL J06YCF(35.0D0, 70.0D0)
          CALL J06YCF(0.0D0, 0.0D0)
          CALL J06YAF(80.0D0, 90.0D0)
          CALL J06YHF('B',1)
          CALL J06YAF(25.0D0, 75.0D0)
          CALL J06YHF('A',1)
          CALL J06YAF(5.0D0,-30.0D0)
          CALL J06YHF('UNCROSSED MECHANISM',19)
C
C
      ENDIF
C
      CALL J06YAF(-10.0D0, -10.0D0)
      CALL J06YHF('O=(0, 0)',8)
      CALL J06YAF(100.0D0, -10.0D0)
      CALL J06YHF('P=(100, 0)',10)
      CALL J06YAF(20.0D0, 0.0D0)
      CALL J06YCF(17.0D0, 2.0D0)
      CALL J06YCF(17.0D0, -2.0D0)
      CALL J06YCF(20.0D0, 0.0D0)
      CALL J06YAF(20.0D0, 2.0D0)
      CALL J06YHF('X',1)
      CALL J06YAF(0.0D0, 0.0D0)
      CALL J06YCF(0.0D0, 20.0D0)
      CALL J06YCF(2.0D0, 17.0D0)
      CALL J06YCF(-2.0D0, 17.0D0)
      CALL J06YCF(0.0D0, 20.0D0)
      CALL J06YAF(-10.0D0, 20.0D0)
      CALL J06YHF('Y',1)
C
      IF (TIBIAL) THEN
          PRINT *,'
          PRINT *,'   WHICH PARAMETER WOULD YOU LIKE TO VARIATE ?
          PRINT *,'
          PRINT *,'    1)  X-coordinate of point A
          PRINT *,'    2)  Y-coordinate ot point A
          PRINT *,'    3)  X-coordinate of point B
          PRINT *,'    4)  Y-coordinate of point B
          PRINT *,'    5)  Slope of straight line
          PRINT *,'    6)  Vertical displacement of straight line
          PRINT *,'
      ELSE
          PRINT *,'
          PRINT *,'   WHICH PARAMETER WOULD YOU LIKE TO VARIATE ?
          PRINT *,'
          PRINT *,'    1)  X-coordinate of point A
          PRINT *,'    2)  Y-coordinate of point A
          PRINT *,'    3)  X-coordinate of point B
          PRINT *,'    4)  Y-coordinate of point B
          PRINT *,'    5)  Convexity/concavity of parabole
          PRINT *,'    6)  Vertical displacement of parabole top
          PRINT *,'    7)  Horizontal displacement of parabole top
```

```
          PRINT *,' ═══════════════════════════════════════════════════════',
        ENDIF
 2      PRINT*, ' Enter number of parameter to be variated'
        READ *, CHAR
        CALL CLEAR
        CALL J06WBF(-25.0D0, 150.0D0, -25.0D0, 150.0D0 ,1)
        IF (TIBIAL) THEN
            IF (CHAR.LT.1.OR.CHAR.GT.6) THEN
                CALL CLEAR
                GOTO 2
            ENDIF
        ELSE
            IF (CHAR.LT.1.OR.CHAR.GT.7) THEN
                CALL CLEAR
                GOTO 2
            ENDIF
        ENDIF
C
C       ***********************************
C       ** VARIATION OF CHOSEN PARAMETER **
C       ***********************************
C
C
        DO 3 I=1,2
C
            J = J + 1
            SA(1,J) = SA(1,1)
            SA(2,J) = SA(2,1)
            SB(1,J) = SB(1,1)
            SB(2,J) = SB(2,1)
            A(J)    = A(1)
            X0(J)   = X0(1)
            H(J)    = H(1)
C
            IF (CHAR.EQ.1) THEN
                SA(1,J) = CC(I) + SA(1,1)
            ELSEIF (CHAR.EQ.2) THEN
                SA(2,J) = CC(I) + SA(2,1)
            ELSEIF (CHAR.EQ.3) THEN
                SB(1,J) = CC(I) + SB(1,1)
            ELSEIF (CHAR.EQ.4) THEN
                SB(2,J) = CC(I) + SB(2,1)
            ELSEIF (CHAR.EQ.5.AND.TIBIAL) THEN
                A(J)        = C(I) + A(1)
            ELSEIF (CHAR.EQ.5) THEN
                A(J)        = CCC(I) + A(1)
            ELSEIF (CHAR.EQ.6) THEN
                H(J)        = CC(I)  + H(1)
            ELSEIF (CHAR.EQ.7) THEN
                X0(J)       = CC(I) + X0(1)
            ENDIF
C
```

```
C
C      ****************************************
C      ** CALCULATION OF EFFECTS OF VARIATION **
C      ****************************************
C
           PRINT *,' Calculating centrodes'
           CALL CENTRO(N, NMAX, J, R, SA, SB, L1, L2, L3, L4,
     *                 TC, FC, PHI)
           PRINT *,' Calculating femoral contact profile'
           CALL FEMORA(N, NMAX, J, A, H, X0, TC, FC, T,
     *                 F, L, TIBIAL, IFAIL, ERROR)
           IF (IFAIL.NE.0) GOTO 4
C
           IF (ERROR.EQ.1) THEN
               PRINT 3000
               PRINT 4000, '                                                '
               PRINT 4000, '  WARNING: It is not possible to                '
               PRINT 4000, '  find a femoral contact profile                '
               PRINT 4000, '  which fits.                                   '
               PRINT 4000, '  Change the vertical displacement              '
               PRINT 4000, '  H (larger) or the convexity                   '
               PRINT 4000, '  A (smaller absolute value).                   '
               PRINT 4000, '  (First look at the results.)                  '
               PRINT 4000, '                                                '
               PRINT 3000
               PRINT *, ' Press ENTER to continue'
               CALL J06WDF
           ENDIF
C
 3     CONTINUE
C
       CALL CLEAR
C
 1000  FORMAT(/,/,/,/,/,/,/,/,/,/)
 2000  FORMAT(19X, 41A)
 3000  FORMAT (/,/,/,/,/,/)
 4000  FORMAT (20X, 40A)
C
 4     END
C
C
C
```

Appendix G: Subroutine SLIP

```
      SUBROUTINE SLIP(N, NMAX, NIN, T, F, VARIAT)
C
C     ******************************************
C     ** THIS ROUTINE CALCULATES THE SLIP RATIO **
C     **          AND PLOTS THE RESULTS         **
C     ******************************************
C
C     ******************************************
C     ** DECLARATION OF THE GLOBAL IDENTIFIERS **
C     ******************************************
C
      INTEGER           N(3), NMAX, NIN
      DOUBLE PRECISION T(2,NMAX,3), F(2,NMAX,3)
      LOGICAL           VARIAT
C
C     ******************************************
C     ** DECLARATION OF THE LOCAL IDENTIFIERS **
C     ******************************************
C
      INTEGER           NNNMAX, J
      PARAMETER         (NNNMAX = 200)
      DOUBLE PRECISION NN(NNNMAX,3)
      DOUBLE PRECISION TL(NNNMAX,3), TDF(NNNMAX,3)
      DOUBLE PRECISION NNMAX, FL(NNNMAX,3), STL(3), SFL(3)
      DOUBLE PRECISION TDFMAX
      DOUBLE PRECISION STDF(3)
C
C     NNNMAX          : MAXIMAL NUMBER OF ITTERATION
C     J               : COUNTER
C     NN(NNNMAX,3)    : FLEXION ANGLE IN DEGREES
C     TL(NNNMAX,3)    : LENGTH OF PART OF TIBIAL CONTACT PROFILE
C     FDF(NNNMAX,3)   : SLIP RATIO
C     NNMAX           :
C     FL(NNNMAX,3)    : LENGTH OF PART OF FEMORAL CONTACT PROFILE
C     STL(3)          : SUM OF TL
C     SFL(3)          : SUM OF FL
C     TDFMAX          : MAXIMAL SLIP RATIO
C     STDF            : AVERAGE SLIP RATIO
C
C     ********************************
C     ** INITIALISATION OF VARIABLES **
C     ********************************
C
      IF(VARIAT) THEN
          NNMAX = MAX(N(1), N(2), N(3))
      ELSE
          NNMAX = N(1)
      ENDIF
C
      J = 0
```

```
      TDFMAX = 0
C
C     ******************************
C     ** CALCULATION OF SLIP RATIO **
C     ******************************
C
 1    J      = J + 1
      STL(J) = 0
      SFL(J) = 0
      DO 2 I = 2, N(J)-2
          TL(I,J)  = SQRT((T(1,I+1,J)-T(1,I,J))*(T(1,I+1,J)-T(1,I,J))+
     *                    (T(2,I+1,J)-T(2,I,J))*(T(2,I+1,J)-T(2,I,J)))
          FL(I,J)  = SQRT((F(1,I+1,J)-F(1,I,J))*(F(1,I+1,J)-F(1,I,J))+
     *                    (F(2,I+1,J)-F(2,I,J))*(F(2,I+1,J)-F(2,I,J)))
          TDF(I,J) = FL(I,J)/TL(I,J)
          TDFMAX   = MAX (TDFMAX, TDF(I, J))
          STL(J)   = STL(J) + TL(I,J)
          SFL(J)   = SFL(J) + FL(I,J)
          NN(I,J)  = 140.0D0*(I-1)/(NIN-1)
 2    CONTINUE
C
C     ***************************************
C     ** CALCULATION OF AVERAGE SLIP RATIO **
C     ***************************************
C
      STDF(J) = SFL(J) / STL(J)
      IF(J.LT.3.AND.VARIAT) GOTO 1
C
C     *************************
C     ** PLOTTING OF RESULTS **
C     *************************
C
      CALL CLEAR
      CALL J06WBF(0.0D0, 140.0D0, 0.0D0, TDFMAX, 1)
      CALL J06WCF(0.0D0, 1.0D0, 0.0D0, 1.0D0)
C
      J = 0
 3    J=J+1
C
C     ***********************************************
C     ** SETTING OF CURRENT LINESTYLE AND COLOUR **
C     ***********************************************
C
C     J06YQF: SETS THE CURRENT COLOUR
C     J06YRF: SETS THE CURRENT LINESTYLE
C
      CALL J06YQF(J)
C
C     **********************
C     ** DRAWING OF LINES **
C     **********************
C
```

```
      CALL J06YAF(NN(2,J),TDF(2,J))
      DO 4 I = 3, N(J)-3
      CALL J06YCF(NN(I,J),TDF(I,J))
 4    CONTINUE
C
C     ************************************
C     ** DRAWING OF AVERAGE SLIP RATIO **
C     ************************************
C
      CALL J06YRF(2)
      CALL J06YAF(NN(2,J),STDF(J))
      CALL J06YCF(NN(N(J)-3,J),STDF(J))
      CALL J06YRF(1)
      IF(J.LT.3.AND.VARIAT) GOTO 3
C
C     ********************************
C     ** DRAWING OF AXES AND TITLES **
C     ********************************
C
      CALL J06YQF(1)
      CALL J06YAF(0.0D0,0.0D0)
      CALL J06YQF(4)
      CALL J06AJF(1,'FLEXION ANGLE')
      CALL J06AJF(2,'SLIP RATIO')
      CALL J06AHF('SLIP RATIO')
      CALL J06YQF(1)
      CALL J06AAF
      CALL J06WDF
      CALL J06WBF(-25.0D0, 150.0D0, -25.0D0, 150.0D0, 1)
      CALL J06YLF(5.0D0,0.0D0)
      CALL J06YKF(5.0D0,7.0D0)
C
      END
C
C
C
```

Appendix H: Subroutine MOVIE

```
      SUBROUTINE MOVIE(N, NMAX, SA, SB, TC, FC, T, F, VARIAT,
     *                 PHI, R, NIN, L4)
C
C     ********************************************************
C     ** THIS SUBROUTINE CALCULATES THE NEW COORDINATES OF **
C     **       SA, SB, FC AND F,  WHEN WE ROTATE L3.        **
C     ********************************************************
C
C     ******************************************
C     ** DECLARATION OF THE GLOBAL IDENTIFIERS **
C     ******************************************
C
      INTEGER           N(3), NMAX, NIN
      DOUBLE PRECISION SA(2,3), SB(2,3), TC(2,NMAX,3), FC(2,NMAX,3)
      DOUBLE PRECISION T(2,NMAX,3), F(2,NMAX,3), PHI(NMAX,3)
      DOUBLE PRECISION R, L4
      LOGICAL           VARIAT
C
C     *****************************************
C     ** DECLARATION OF THE LOCAL IDENTIFIERS **
C     *****************************************
C
      INTEGER           I, II, III, J, NNMAX, NUM, CHAR
      PARAMETER         (NNMAX = 200, NUM = 7)
      DOUBLE PRECISION FCLO(2,NNMAX,3), FLO(2,NNMAX,3), SBLO(2,3)
      DOUBLE PRECISION ROT, TRANS(2), NESA(2,3,NUM), NESB(2,3,NUM)
      DOUBLE PRECISION NEFC(2,NNMAX,3,NUM), NEF(2,NNMAX,3,NUM), PI
      DOUBLE PRECISION SALPHA, L1
      PARAMETER         (PI = 3.141592653)
C
C     I              : COUNTER
C     II             : COUNTER
C     III            : COUNTER
C     J              : COUNTER
C     NNNMAX         : MAXIMAL NUMBER OF ITERATIONS
C     NUM            : COUNTER (NUMBER OF MOVING STEPS)
C     CHAR           : INPUT VARIABLE
C     FCLO(2,NMAX,3) : LOCAL COORDINATES OF FEMORAL CENTRODE (FC)
C     FLO(2,NMAX,3)  : LOCAL COORDINATES OF FEMORAL CONTACT PROFILE (F)
C     SBLO(2,3)      : LOCAL COORDINATES OF B
C     ROT            : ROTATION ANGLE
C     TRANS(2)       : TRANSLATION VECTOR
C     NESA(2,3,NUM)  : COORDINATES OF A AFTER TRANSLATION/ROTATION
C     NESB(2,3,NUM)  : COORDINATES OF B AFTER TRANSLATION/ROTATION
C     NEFC(2,NMAX,3,NUM) :COORDINATES OF FC AFTER TRANSLATION/ROTATION
C     NEF(2,NMAX,3,NUM) : COORDINATES OF F AFTER TRANSLATION/ROTATION
C     PI             : PI = 3.141592653
C     SALPHA         : ALPHA AT THE GIVEN POSITION
C     L1             : LENGHTS OF LINK OA
C
```

```
C
      J = 1
C
 15   IF(VARIAT) THEN
        PRINT *,'
        PRINT *,'   WHICH MECHANISM WOULD YOU LIKE TO MOVE ?
        PRINT *,'
        PRINT *,'     1)  The mechanism without   parameter variation
        PRINT *,'     2)  The mechanism with neg. parameter variation
        PRINT *,'     3)  The mechanism with pos. parameter variation
        PRINT *,'     4)  Back to the main menu
        PRINT *,'
        PRINT*, ' Enter number of your choice'
        READ *, CHAR
        CALL CLEAR
        IF(CHAR.EQ.1) THEN
            J = 1
        ELSEIF(CHAR.EQ.2) THEN
            J = 2
        ELSEIF(CHAR.EQ.3) THEN
            J = 3
        ELSEIF(CHAR.EQ.4) THEN
            GOTO 16
        ELSE
            GOTO 15
        ENDIF
      ENDIF
C
C     ******************************************
C     ** CALCULATION OF THE LOCAL COORDINATES **
C     ******************************************
C
      DO 2 I = 2, N(J)-1
          FCLO(1,I,J) = FC(1,I,J) - SA(1,J)
          FCLO(2,I,J) = FC(2,I,J) - SA(2,J)
          FLO(1,I,J)  = F(1,I,J)  - SA(1,J)
          FLO(2,I,J)  = F(2,I,J)  - SA(2,J)
          SBLO(1,J)   = SB(1,J)   - SA(1,J)
          SBLO(2,J)   = SB(2,J)   - SA(2,J)
 2    CONTINUE
C
C     ******************************************
C     ** CALCULATION OF THE TRANSLATION VECTOR **
C     **        AND THE ROTATION ANGLE          **
C     ******************************************
C
      SALPHA = R + ATAN ((SB(2,J)-SA(2,J))/(SB(1,J)-SA(1,J)))
      L1 = SQRT(SA(1,J)*SA(1,J) + SA(2,J)*SA(2,J))
C
      DO 3 II = 1, NUM
          III           = 1.0D0*(II-1)/(NUM-1)*(N(J)-3) + 2
          ROT           = ((50.0D0/180.0D0)*PI) + R -
```

```
     *                    (((III-1.0D0)/(NIN-1.0D0))*
     *                    (140.0D0/180.0D0)*PI) - SALPHA
           NESA(1,J,II)  = L1 * COS(PHI(III,J))
           NESA(2,J,II)  = L1 * SIN(PHI(III,J))
           TRANS(1)      = NESA(1,J,II) - SA(1,J)
           TRANS(2)      = NESA(2,J,II) - SA(2,J)
C
C          ************************************************
C          ** CALCULATION OF THE NEW COORDINATES OF B,     **
C          ** AND THE FEMORAL CENTRODE AND CONTACT PROFILE **
C          ************************************************
C
           NESB(1,J,II)  = SA(1,J) + TRANS(1)
     *                   + SBLO(1,J)*COS(ROT)
     *                   - SBLO(2,J)*SIN(ROT)
           NESB(2,J,II)  = SA(2,J) + TRANS(2)
     *                   + SBLO(1,J)*SIN(ROT)
     *                   + SBLO(2,J)*COS(ROT)
C
           DO 4 I = 2, N(J)-1
               NEFC(1,I,J,II) = SA(1,J) + TRANS(1)
     *                        + FCLO(1,I,J)*COS(ROT)
     *                        - FCLO(2,I,J)*SIN(ROT)
               NEFC(2,I,J,II) = SA(2,J) + TRANS(2)
     *                        + FCLO(1,I,J)*SIN(ROT)
     *                        + FCLO(2,I,J)*COS(ROT)
               NEF(1,I,J,II)  = SA(1,J) + TRANS(1)
     *                        + FLO(1,I,J)*COS(ROT)
     *                        - FLO(2,I,J)*SIN(ROT)
               NEF(2,I,J,II)  = SA(2,J) + TRANS(2)
     *                        + FLO(1,I,J)*SIN(ROT)
     *                        + FLO(2,I,J)*COS(ROT)
 4         CONTINUE
 3     CONTINUE
C
 17    PRINT *,'
       PRINT *,'   WHICH TYPE OF PLOT ?                    '
       PRINT *,'
       PRINT *,'   1)  Several positions in one plot       '
       PRINT *,'   2)  More positions one by one           '
       PRINT *,'   3)  Back to the main menu               '
       PRINT *,'
       PRINT*, ' Enter number of your choice'
       READ *, CHAR
       CALL CLEAR
       IF(CHAR.EQ.1) THEN
           GOTO 18
       ELSEIF(CHAR.EQ.2) THEN
           GOTO 19
       ELSEIF(CHAR.EQ.3) THEN
           GOTO 16
       ELSE
```

```
          GOTO 17
        ENDIF
C
C       ***************************************************
C       ***************************************************
C       **** DRAWING 1/3*NUM MECHANISMS IN ONE PICTURE ****
C       ***************************************************
C       ***************************************************
C
C       ********************************
C       ** DRAWING OF TIBIAL CENTRODE **
C       ********************************
C
C       J06YAF:    MOVES PEN TO A GIVEN POINT
C       J06YCF:    DRAWS LINE BETWEEN TWO POINTS
C
 18     CALL J06YQF(1)
        CALL J06YAF(TC(1,2,J),TC(2,2,J))
        DO 5 I=3, N(J)-1
            CALL J06YCF(TC(1,I,J),TC(2,I,J))
 5      CONTINUE
C
C       ***************************************
C       ** DRAWING OF TIBIAL CONTACT PROFILE **
C       ***************************************
C
        CALL J06YQF(3)
        CALL J06YAF(T(1,2,J),T(2,2,J))
        DO 6 I=3,N(J)-1
            CALL J06YCF(T(1,I,J),T(2,I,J))
 6      CONTINUE
C
        DO 9 II = 1, NUM ,3
C
C           ***********************************
C           ** DRAWING OF FEMORAL CENTRODES **
C           ***********************************
C
            CALL J06YQF(2)
            CALL J06YAF(NEFC(1,2,J,II),NEFC(2,2,J,II))
            DO 7 I=3,N(J)-1
                CALL J06YCF(NEFC(1,I,J,II),NEFC(2,I,J,II))
 7          CONTINUE
C
C           ****************************************
C           ** DRAWING OF FEMORAL CONTACT PROFILES **
C           ****************************************
C
            CALL J06YQF(4)
            CALL J06YAF(NEF(1,2,J,II),NEF(2,2,J,II))
            DO 8 I=3,N(J)-1
                CALL J06YCF(NEF(1,I,J,II),NEF(2,I,J,II))
```

```
 8          CONTINUE
C
C           ****************************
C           ** DRAWING THE FOUR LINKS **
C           ****************************
C
            CALL J06YQF(5)
            CALL J06YAF(0.0D0, 0.0D0)
            CALL J06YCF(L4, 0.0D0)
            CALL J06YCF(NESB(1,J,II), NESB(2,J,II))
            CALL J06YCF(NESA(1,J,II), NESA(2,J,II))
            CALL J06YCF(0.0D0, 0.0D0)
 9      CONTINUE
C
        PRINT 1000
        PRINT *, 'Press enter to continue'
C
        CALL J06WDF
        CALL J06YQF(1)
        IF(VARIAT) GOTO 15
        GOTO 17
C
C       ******************************************
C       ******************************************
C       **** DRAWING NUM MECHANISMS ONE BY ONE ****
C       ******************************************
C       ******************************************
C
 19     DO 10 II=1, NUM
C
C           ********************************
C           ** DRAWING OF TIBIAL CENTRODE **
C           ********************************
C
C           J06YAF:    MOVES PEN TO A GIVEN POINT
C           J06YCF:    DRAWS LINE BETWEEN TWO POINTS
C
            CALL J06YQF(1)
            CALL J06YAF(TC(1,2,J),TC(2,2,J))
            DO 11 I=3, N(J)-1
                CALL J06YCF(TC(1,I,J),TC(2,I,J))
 11         CONTINUE
C
C           ****************************************
C           ** DRAWING OF TIBIAL CONTACT PROFILE **
C           ****************************************
C
            CALL J06YQF(3)
            CALL J06YAF(T(1,2,J),T(2,2,J))
            DO 12 I=3,N(J)-1
                CALL J06YCF(T(1,I,J),T(2,I,J))
 12         CONTINUE
```

```
C
C
C              *********************************
C              ** DRAWING OF FEMORAL CENTRODE **
C              *********************************
C
               CALL J06YRF(3)
               CALL J06YQF(2)
               CALL J06YAF(NEFC(1,2,J,II),NEFC(2,2,J,II))
               DO 13 I=3,N(J)-1
                   CALL J06YCF(NEFC(1,I,J,II),NEFC(2,I,J,II))
 13            CONTINUE
C
C              ***************************************
C              ** DRAWING OF FEMORAL CONTACT PROFILE **
C              ***************************************
C
               CALL J06YQF(4)
               CALL J06YAF(NEF(1,2,J,II),NEF(2,2,J,II))
               DO 14 I=3,N(J)-1
                   CALL J06YCF(NEF(1,I,J,II),NEF(2,I,J,II))
 14            CONTINUE
C
C          ****************************
C          ** DRAWING THE FOUR LINKS **
C          ****************************
C
           CALL J06YRF(1)
           CALL J06YQF(5)
           CALL J06YAF(0.0D0, 0.0D0)
           CALL J06YCF(L4, 0.0D0)
           CALL J06YCF(NESB(1,J,II), NESB(2,J,II))
           CALL J06YCF(NESA(1,J,II), NESA(2,J,II))
           CALL J06YCF(0.0D0, 0.0D0)
C
      PRINT 1000
      PRINT *, 'Press enter to continue'
      CALL J06WDF
 10   CONTINUE
C
      CALL J06YQF(1)
      IF(VARIAT) GOTO 15
      GOTO 17
C
 1000 FORMAT(/,/,/,/,/,/,/,/,/,/,/,/,/,/,/,/,/,/,/,/,/,/,/,/,/,/,/)
 16   END
```

USER'S MANUAL

Report Number 92.013 C

Frank Peters
Niek van Nunen

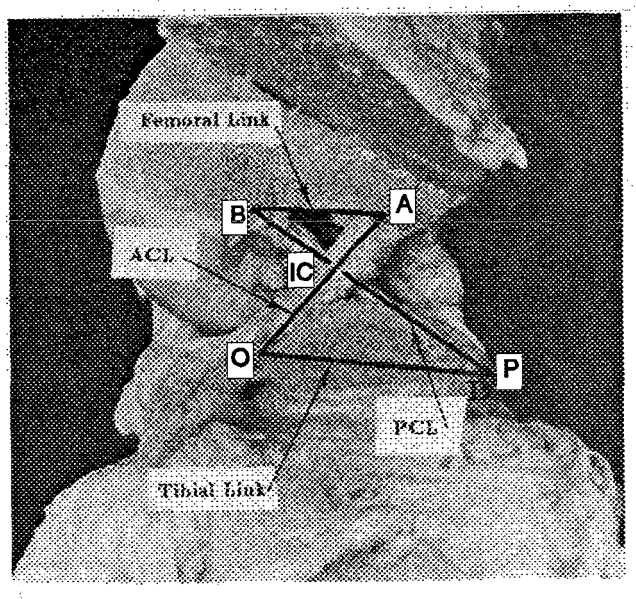
Eindhoven University of Technology, January 1992

1. Contents

1. Contents **1**
2. Introduction **2**
3. The program **3**
3.1 Starting the program 3
3.2 The main menu 3
3.2.1 Input of a mechanism 4
3.2.1.1 Anatomic dimensions 5
3.2.1.2 Perspex model 5
3.2.1.3 Self made mechanism 6
3.2.2 Plot results 6
3.2.3 Parameter variation 7
3.2.4 Slip ratio 8
3.2.5 Movie 9
3.2.6 Quit 10
3.3 The calculations 10
3.4 Termination of the program 12
4. Finally **12**
5. Literature **13**

2. Introduction

The bones that meet a joint, as well as the ligaments that hold the bones together, can be analyzed as a mechanical linkage. Since the movements allowed to the bones at the human knee occur mainly in the sagittal plane, much can be learned by treating the knee as a two-dimensional single-degree-of-freedom linkage, moving in a single plane. *figure 1* shows a human knee from which the lateral femoral condyle has been excisted, exposing the cruciate ligaments. The ligaments, together with the two bones, form the cruciate linkage. *AB* and *PB* represent the anterior (ACL) and posterior (PCL) cruciate ligaments, respectively.



The program knee simulates a 2 dimensional kinematic model of a knee joint. It calculates, given a mechanical linkage and tibial contact profile, the corresponding tibial and femoral centrodes and the femoral contact profile. A normal knee joint can move from full extension to about 140° flexion. These values will also be used in the program. The user has the opportunity to select a straight line as tibial contact profile or a parabole. He can also choose between a crossed or an uncrossed linkage. After the program has calculated the centrodes and contact profiles, the user can variate one of the parameters and examine the effects of the parameter variation. The slip ratio between the tibial and the femoral contact profile can be plotted in a diagram. The program also has the possibility to plot the mechanism with the centrodes and contact profiles in several positions for different flexion angles. All the calculations will be made for a mechanism with rigid elements. It's the intention to extend the program with the opportunity to use flexible elements, but at this moment the program does not have this opportunity yet.

3. The program

3.1 Starting the program

It's easy to start the program. When the disk is in the disk drive, the user only has to type:

A: [ENTER]
knee [ENTER]

As soon the program has started it is not allowed to use strings anymore (the easiest way to avoid that this will happen anyhow, is to use only the numerical part of the keyboard), otherwise the program will be terminated. First the program will ask which graphical device is to be used. **Always choose: 1. PC SCREEN.** This switches the terminal to the graphical mode.

3.2 The main menu

Next the main menu will appear (*figure 2*).

```
MAIN MENU

 1) Input of a mechanism
 2) Plot results
 3) Parameter variation
 4) Slip ratio
 5) Movie
 6) Quit

Enter number of your choice
```

figuur 2: The main menu

When no mechanism has been chosen yet, the user can only choose: 1) Input of a mechanism. Otherwise a message "No mechanism chosen yet" will be given. As soon as a mechanism has been chosen, also one of the other opportunities can be chosen.

3.2.1 Input of a mechanism

The program needs the following parameters (*figure 3*):

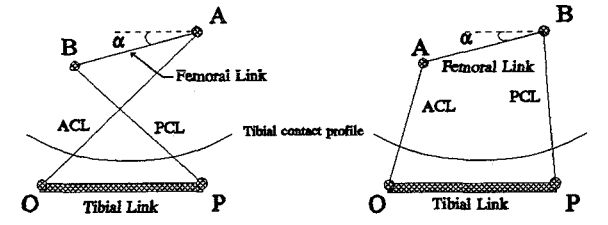


Figure 3: The two types of mechanisms and some of the parameters to be entered.

- Whether the mechanism is a crossed or an uncrossed one.
- The coordinates of the points A and B (the coordinates of the points O and P are not variable: $O = (0, 0)$ and $P = (100, 0)$.
- Whether the tibial contact profile will be described by a straight line or a parabole.
- The parameters of the straight line or the parabole:
  - The straight line will be described by:
    " $Y = AX + H$ "
    with A the slope of the straight line and.
    H the heigth of the line at $X = 0$.
  - The parabole will be described by:
    " $Y = A(X-X_0)^2 + H$ "
    with A the convexity/concavity of the parabole.
    X_0 the x-coordinate of the top of the parabole.
    H the heigth of the top of the parabole.
  - The number of increments (N).

When the user chooses option 1 at the main menu, the choice out of three kinds of mechanisms will be given (see *figure 4*).

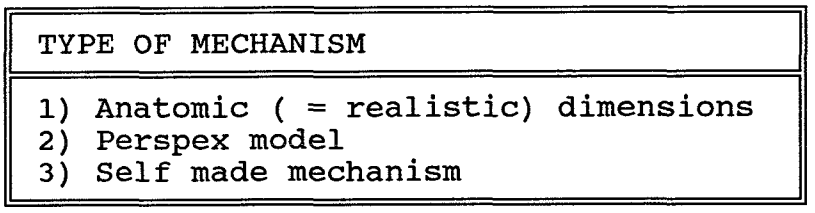


Figure 4: The menu which appears when "1) Input of a mechanism" is chosen.

3.2.1.1 Anatomic dimensions

The parameters for a mechanism with anatomic (realistic) dimensions are stored in the program. these parameters are:

* *The mechanism is a crossed one*
* $A = (61, 77)$
* $B = (20, 68)$
* *The tibial profile is a parabole*
* $A = -0.005$
* $X_0 = 50$
* $H = 7$
* $N = 200$

3.2.1.2 Perspex model

The faculty of Medical Engineering Technology of the Eindhoven University of Technology has the disposal of a 2 dimensional Perspex model of the knee joint. The dimensions of this model are also stored in the program. These values for the parameters then are:

* *The mechanism is a crossed one*
* $A = (91, 78)$
* $B = (42, 78)$
* *The tibial profile is a straight line*
* $A = 0$
* $H = 7$
* $N = 200$

3.2.1.3 Self made mechanism

In this case the user has to enter the parameters him/herself. The program will ask for them one by one. After every value the user gives, the ENTER-button has to be pressed. The program will also give a recommended range for some of the parameters.

3.2.2 Plot results

This option draws a plot of the mechanism in the given position. It is possible that there will be no contact between the tibial contact profile and the femoral contact profile. In this case the given position has a flexion angle which is less than 0° or larger than 140°. The program only calculates for a range between 0° and 140°.
In the case that the user didn't use the opportunity of parameter variation yet, there will be drawn only one mechanism:

- The **yellow** line is the input situation of the linkage.
- The **white** line is the tibial centrode.
- The **red** line is the femoral centrode.
- The **green** line is the tibial contact profile.
- The **blue** line is the femoral contact profile.

In the other case (the option parameter variation has been chosen yet) there will be drawn three mechanisms:

- The **white** mechanism is the one without parameter variation.
- The **red** mechanism is the one with negative parameter variation.
- The **green** mechanism is the one with positive parameter variation.
  (see also 3.2.3 Parameter variation)

3.2.3 Parameter variation

When the user chooses the option "3) parameter variation", the next menu will appear (*figure 5*):

```
PARAMETER VARIATION

1) Elements are rigid
2) Elements are flexible

Enter number of your choice
```

figure 5 The parameter variation menu. The choise for flexible elements is not yet possible.

The user has to choose for opportunity 1, while opportunity 2 is not available yet. In the situation where the tibial contact profile is a straight line, the following menu will appear (*figure 6*):

```
WHICH PARAMETER WOULD YOU LIKE TO VARIATE ?

1) X-coordinate of point A
2) Y-coordinate of point A
3) X-coordinate of point B
4) Y-coordinate of point B
5) Slope of straight line
6) Vertical displacement of straight line

Enter number of parameter to be variated
```

Figure 6: Menu for parameter variation with tibial contact profile is a straight line.

When the user chooses for options 1, 2, 3, 4 or 6 the chosen parameter will set on:

- Its old value - 10
- Its old value + 10

When the user chooses for option 5 the slope of the line will be changed in:

- Its old value - 0.1
- Its old value + 0.1

The program will calculate the centrodes and the contact profiles for both cases.

If a parabole is chosen for the tibial contact profile the user will get the next menu (*figure 7*):

```
WHICH PARAMETER WOULD YOU LIKE TO VARIATE ?

1) X-coordinate of point A
2) Y-coordinate of point A
3) X-coordinate of point B
4) Y-coordinate of point B
5) Convexity/concavity of parabole
6) Vertical displacement of parabole top
7) Horizontal displacement of parabole top

Enter number of parameter to be variated
```

Figure 7: Menu for parameter variation with femoral contact profile is a parabole.

When the user choses for options 1, 2, 3, 4, 6 or 7 the choosen parameter will be changed in:

- Its old value -10
- Its old value +10

When option 5 is chosen the convexity/concavity of the parabole will be changed in:

- Its old value -0.001
- Its old value +0.001

3.2.4 Slip ratio

This option calculates the slip ratio and plots it against the flexion angle. When the slip ratio is equal to 1 this means there is no slip. In the diagram the average slip is also plotted. When the option parameter variation has been chosen three lines will be drawn:

- The **white** one for the mechanism without variation.

- The **red** one for the mechanism with neg. variation.
- The **green** one for the mechanism with pos. variation.

3.2.5 Movie

The program will calculate a number of different positions of the femur with respect to the tibia. If the option parameter variation has been chosen, the next menu will appear (*figure 8*):

```
WHICH MECHANISM WOULD YOU LIKE TO MOVE ?

1) The mechanism without parameter variation
2) The mechanism with neg. parameter variation
3) The mechanism with pos. parameter variation
4) Back to the main menu

Enter number of your choice
```

Figure 8: Option menu for routine MOVIE (this menu only appears when the option parameter variation has been chosen yet.

In this way each of the three possible mechanisms can be shown.
After the user has chosen a mechanism, and if the option parameter variation has not been chosen yet, the following menu will appear (see *figure 9*):

```
WHICH TYPE OF PLOT ?

1) Several positions in one plot
2) More positions one by one
3) Back to the main menu

Enter number of your choice
```

Figure 9: The option MOVIE

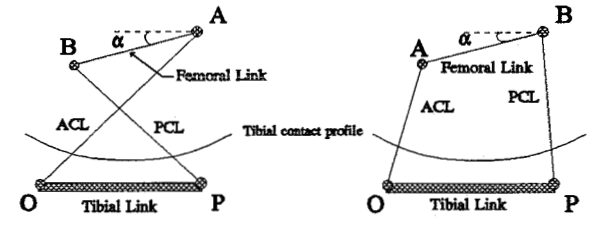
The user can choose between several positions of the mechanism in one and the same plot, or he can look at several calculated positions of the mechanisms one by one. In this case more plots will be drawn than in the first case.

3.2.6 Quit

This option will quit the program. It also resets the terminal to the normal (VGA/EGA/....) video mode.

3.3 The calculations

When the user has given a mechanism, or when he has variated one of the parameters, the program will start the calculations automatically. The centrodes and the profiles will be calculated from full extension (flexion angle is 0°) to full flexion (flexion angle is 140°). This agrees with a range of -90° to 50° for the angle α of the femoral link (see *figure 10*).



a. crossed mechanism b. uncrossed mechanism

In some cases the program will give a warning:

1) When the program rotates the tibial link from -90° to 50° it is possible that the tibial link becomes in a direct line with one of the cruciate ligaments. When this happens, the femoral centrode will get what is called a "tail" (see *figure 11*).

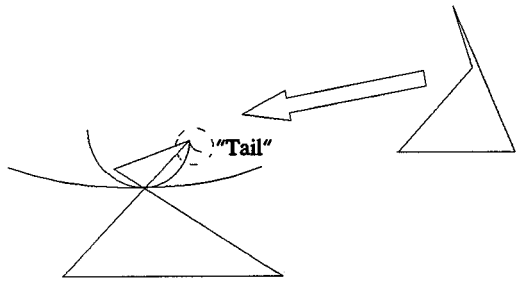


Figure 11: Demonstration of the problem that appears when the mechanism changes from crossed to uncrossed. The program will detect the beginning of the "Tail" and stop further calculations of thecentrode.

T h i s
causes a discontinuity of the femoral contact profile. To avoid this problem the program checks if this is happening and stops the calculations at that moment. Then the user will get the following warning (*figure 12*):

```
It is not possible to calculate the full range
of the flexion angle (gamma) for this system.
The program will calculate the centrodes and
the femoral contact profile for:

              0° ≤ gamma ≤  .....°

The mechanism will be drawn in .......
```

Figure 12: The warning given when it is not possible to simulate the whole flexion range.

The user can read in the warning for which range of the flexion angle the program could calculate the centrodes. The warning gives also the colour of the plot this mechanism will be drawn in. This is only of interest when the option parameter variation has been chosen.

2) It is possible that the combination of the chosen linkage and the tibial contact profile does not fit. In that case the next warning will be given (*figure 13*):

```
WARNING: It is not possible to
find a femoral contact profile
which fits.
Change the vertical displacement
H (larger) or the convexity
A (smaller absolute value).
(First look at the results.)
```

Figure 13: Warning when it is not possible to calculate a combination of contact profiles that fit. Also some suggestions to solve the problem are given.

The program doesn't stop the calculations, so it's possible to take a look to what happens in this case. Sometimes it is possible to find parts of the profiles which could fit but it is impossible to find a femoral contact profile which fits for the whole flexion range.

3.4 Termination of the program

When the program is terminated, for example when a string is entered where a number was expected, the monitor will not be reset to the normal (VGA/EGA/....) video mode. The easiest way to get back to this mode is the restart the program (by typing knee [ENTER]). Now the user can continue the program, or if desired quit the program from the main menu.

4. Finally

This program has been written for a simulation practical that is a part of the compulsory part of the curriculum of the faculty of Medical Engineering Technology of the Eindhoven University of Technology. More information about the modelling of the knee by a two dimensional model, is given during one of the courses, Functional anatomy. More information about the model can also be derived from Lit. [1], [2] and [5] (see *5. Literature*).

5. Literature

[1] Ian A.F. Stokes (Editor)
Mechanical factors and the skeleton
John Libbey & Company Limited, London
ISBN 0-86196-006-8

[2] Dale Daniel, Wayne Akeson, John O' Connor
Knee Ligaments; Structure, Function, Injury and Repair
Raven Press, New York
ISBN 0-88167-605-5

[3] Richard S. Hartenberg and Jacques Denavit
Kinematic Synthesis of Linkages
McGraw-Hill series in MECHANICAL ENGINEERING
McGraw-Hill Book company, 1964

[4] dr. G.R. Veldkamp
Kinematica
Scheltema & Holkema, Amsterdam, 1970
ISBN 90-6060-502-0

[5] A. Huson, C.W. Spoor and A.J. Verbout
A Model of the Human Knee, Derived from Kinematic Principles and Its Relevance for Endoprothesis Design.
Acta Morphologica Neerlando-Scandinavica 27 (1989), pp. 45-62

[6] Software used:

Wordperfect 5.1 NL
Drawperfect 1.1 NL
MS-Fortran 5.0 Compiler
NAG Workstation Library
NAG PC Graphics library
Norton Classic Editor, version 1.5

All software used was retrieved from the Calculation Center (Rekencentrum) of the Eindhoven University of Technology.

[7] Hardware used:

Rembrant 386 AT with 80387 coprocessor
VGA card with colour monitor

This hardware was retrieved from the Department of Fundamental Research, Faculty of Mechancal Engineering of the Eindhoven University of Technology